P. 1877/96

Wybór systemu MRP II

miesięcznik profesjonalistów

12
1996

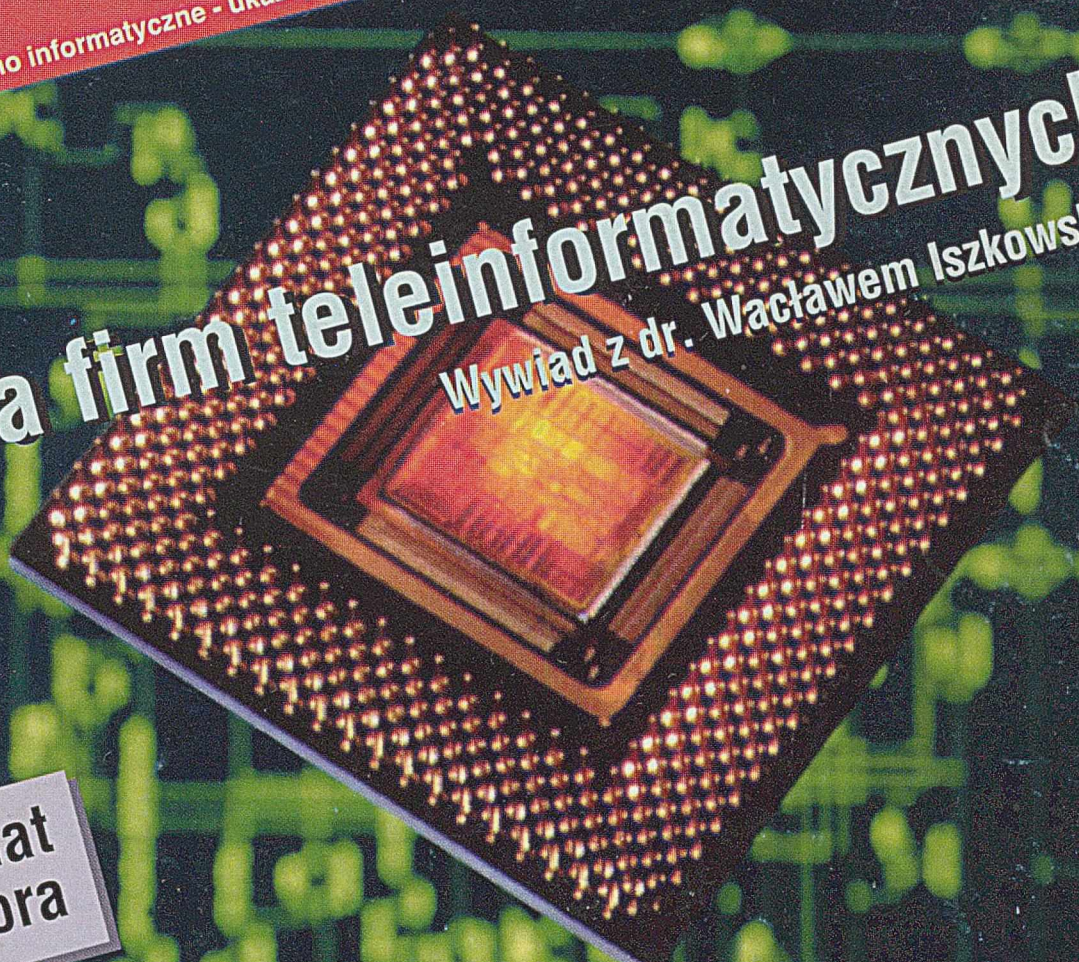
informatyka

Cena: 5 zł. (50 000 zł.) ISSN 0542 9951 WYDAWNICTWO SIGMA NOT

Pismo informatyczne - ukazuje się od 1965 roku

Izba firm teleinformatycznych

Wywiad z dr. Wacławem Iszkowskim



25 lat mikroprocesora

Algorytmy genetyczne

Wesołych Świąt

Szczęśliwego Nowego

Roku

informatyka

Miesięcznik
ISSN 0542-9951
INDEKS 36124

REDAGUJE ZESPÓŁ:

dr Lesław **WAWRZONEK**
(redaktor naczelny)
mgr Jerzy **SZYLLER**
(z-ca redaktora naczelnego)
mgr Zdzisław **ŻURAKOWSKI**
(redaktor działu)
Agnieszka **PAPIS**
(sekretarz redakcji)
Alina **KLEPACZ**
(administracja redakcji)

KOLEGIUM REDAKCYJNE:
prof.dr hab. Leonard BOLC
mgr inż. Piotr FUGLEWICZ
prof. dr hab. Jan GOLIŃSKI
dr inż. Zenon KULPA
prof. dr inż. Jan MULAWKA
prof. dr hab. Wojciech OLEJNICZAK
mgr. inż. Jan RYŻKO
dr Witold STANISZKIS
dr inż. Jacek STOCHLAK
prof. dr hab. Maciej STOLARSKI
prof. dr hab. Zdzisław SZYJEWSKI
prof. dr hab. inż.
Ryszard TADEUSIEWICZ
prof. dr hab. Jan WĘGLARZ

PRZEWODNICZĄCY
RADY PROGRAMOWEJ
prof. dr hab. Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek
Technicznych SIGMA NOT Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

REDAKCJA:

00-950 Warszawa,
ul. Ratuszowa 11, p. 628
skrytka pocztowa 1004
tel., fax: 619-11-61
tel.: 619-22-41 w. 159
e-mail: informat@pol.pl

Materiałów nie zamówionych redakcja nie zwraca.

Autorzy artykułów proszeni są o przysyłanie tekstów na dyskietkach 31/2" - w czystych kodach ASCII (Latin II, Mazovia) lub edytorach: Word 2, Word 6.
Po szczegółowe informacje dla Autorów prosimy zwracać się do redakcji.

Redakcja nie ingeruje w treść i formę ogłoszeń i innych materiałów reklamowych, w związku z tym nie ponosi za nie odpowiedzialności.

Ogłoszenia przyjmują:
- Redakcja, tel. 619-11-61
- Dział Reklamy i Marketingu
00-950 Warszawa, ul. Mazowiecka 12
tel.: 27-43-66, fax: 26-80-16

Okładka:
AGAT, Jerzy Burski i Andrzej Jacyszyn,
zdjęcie: Digital Equipment
Łamanie:
Alina Klepacz, program PageMaker 5.0
Druk: Drukarnia SIGMA NOT Sp. z o.o.

UWAGA
Ukazuje się już elektroniczna wersja naszego miesięcznika. Oto nasz adres w Internecie:
http://www.pol.pl/informatyka/

nr 12 1996
Grudzień
Rok wydania XXXII

W numerze:

P1877/96

Strona

Informacje 2

Wywiad

Izba firm teleinformatycznych 8
Wywiad z dr. Wacławem Iszkowskim

Systemy informacyjne

Rozważnie, ale i szybko – czyli rzecz o tym, jak wybierać system informatyczny wspomagający MRPII 9
Aleksander Popończyk

Jak budować system Computer Integrated Manufacturing? Technologia 13
Józef B. Lewoc, Antoni Sydor

Publikacje

Oprogramowanie do optymalizacji wykorzystujące Algorytmy Genetyczne 17
Dariusz Misiurek

Systemy czasu rzeczywistego 23
Język programowania CHILL
Marek Średniawa, Krzysztof Brzeziński

Terminologia

Język Java i komputer sieciowy... 31
Jerzy Szyller

OPINIE

25 lat mikroprocesora czyli po co jemy tę żabę 32
Jerzy Szych

O mitologii komputeryzacji moje trzy grosze 33
Stefan Sokołowski

zBITki

Wznowienie 36
Wacław Iszkowski

Ochrona antywirusowa

INITEL-Serwis, firma zajmująca się dystrybucją produktów pamięci masowych, od niedawna proponuje swoim klientom oprogramowanie antywirusowe, podnoszące bezpieczeństwo pracy w systemie Windows NT, VirusUtilities austriackiej firmy IKARUS. Oprogramowanie to nie jest klasycznym "skanerem", lecz pracuje jako tzw. "NT-Serwis" w tle. Oznacza to, że poza bieżącą kontrolą stale chroni serwer, również w sytuacji, gdy żaden użytkownik nie jest z nim połączony. (ap)

California Computer dystrybutorem produktów Fujitsu w Polsce

Firma California Computer podpisała umowę dystrybucyjną z firmą Fujitsu NordicAB. California Computer ma sprzedawać dyski twarde Fujitsu do ponad 1500 swoich dealer'ów i reseler'ów. Fujitsu jest obecnie jednym z czołowych producentów dysków twardych na świecie, a *"California Computer jest firmą, przy której udziale Fujitsu pragnie osiągnąć wiodącą pozycję na rynku dysków twardych w Polsce"*, jak powiedział Arne Wahlberg, Sales 7 Marketing Director w Fujitsu Nordic. (ap)

Programy multimedialne firmy Microsoft

Podczas Targów Gier Komputerowych **Gambleriada'96** firma Microsoft zaprezentowała nowe programy multimedialne, służące do ułatwienia i uprzyjemnienia procesu nauczania. Przede wszystkim są to encyklopedie multimedialne o różnorodnej tematyce, adresowane zarówno do dzieci, jak i dorosłych. Zaprezentowano m.in.: **Microsoft Encarta® 97** Encyclopedia, zawierającą 30 000 haseł, wzbogaconych grafiką, zdjęciami, animacjami i sekwencjami video; **Microsoft Encarta® World Atlas**, który zawiera mapy kontynentów, poszczególnych krajów, a nawet plany największych miast świata, wzbogacony o zdjęcia satelitarne, **Microsoft Cinemania® 97** z najnowszymi informacjami o świecie kina, itd. Wszystkie encyklopedie są co miesiąc aktualizowane, a odpowiednie pliki można znaleźć na internetowych stronach Microsoftu: **http:/w.microsoft.com/poland/**.

Przedstawiona została też seria **Scholastic's The Magic School Bus,** przeznaczona dla dzieci w wieku od 6 do 10 lat. Składa się ona z pięciu pozycji, które łączą funkcje edukacyjne z zabawą. Każdy program zajmuje się poszczególnym zagadnieniem, a wiedzę utrwala się rozwiązując zagadki i łamigłówki.

Microsoft oferuje także inne produkty, które przeznaczone są dla trochę starszych dzieci, jak np. **3D Movie Maker**, przy pomocy którego można samemu stworzyć film animowany. (ap)

Acer i Texas Instruments zawarły porozumienie o wymianie licencji

Acer Group, siódmy na świecie producent komputerów osobistych i amerykańska Texas Instruments zawarły porozumienie o wzajemnej wymianie patentów obejmującej technologie produkcji komputerów osobistych, urządzeń peryferyjnych oraz monitorów. Umowa ta umożliwi obu stronom rozwijanie nowych technologii w dziedzinie sprzętu PC. (ap)

Microsoft i Parker Software opracują programy-klientów unixowych do SNA Servera

Firmy Microsoft Corp. oraz Parker Software Inc. opracowały rodzimy program-klient UNIX'a do Microsoft® SNA Servera. Microsoft oferuje podobne programy-klientow do systemów operacyjnych MS-DOS®, Windows® 3.x, Windows for Workgroups, OS/2, Windows 95 i Windows NT® Workstation. Umożliwiają one połączenie poprzez SNA Server stacji roboczych sieci lokalnej z komputerami mainfraime IBM'u i systemami AS/400. Parker Software udostępni unixowe programy-klientów do wszystkich popularnych wariantów UNIX'a. (ap)

Zarządzanie za pomocą pojedynczej konsoli

Firma Compaq Computer wprowadziła na światowy rynek Compaq Insight Manager do HP Open View. Firma zaczęła realizację strategii, zmierzającej do scentralizowanego zarządzania systemami Compaqa w całym przedsiębiorstwie za pomocą aplikacji preferowanej przez klienta. Compaq Insight Manager do HP Open View jest to unixowe narzędzie do zarządzania serwerem. Pozwala ono użytkownikom na bieżąco śledzić status serwerów i komputerów osobistych Compaqa w całym przedsiębiorstwie za pomocą centralnej konsoli, na której działa Hewlett Packard Open View do systemu HP-UX 3.3.

W pełni funkcjonalna wersja pokazowa Compaq Insight Manager'a do Open View, która kontroluje maksymalnie pięć produktów Compaqa, jest dostępna w ośrodku WWW Compaqa pod adresem **http://www.compaq.com.** (ap)

Nowy notebook AcerNote Light

Nowy notebook firmy Acer - AcerNote Light jest wyposażony w procesor Intel Pentium 100 MHz, 8 MB pamięci EDO RAM oraz 64-bitową szynę pamięci. Notebook ma wymiary 297x215x47 mm i waży 2,65 kg z bateriami i twardym dyskiem. Ma on także magistralę typu PCI. Główna pamięć systemowa może być rozszerzona do 40 MB. Wydajność systemu została podniesiona dzięki zainstalowaniu 16 KB wewnętrznej pamięci typu cache. (ap)

Ułatwienie dla osób niedowidzących

Outlook firmy Xerox jest niedrogim elektronicznym systemem powiększania obrazu, który pomaga czytać osobom niedowidzącym. Przy użyciu systemu Outlook nawet drobne litery w gazetach mogą być powiększone do wysokości ok. 8 cm i wyświetlone na monitorze o wysokim kontraście. Outlook ma wymiary telewizora, waży ok. 15 kg i posiada ruchomy pulpit, który ułatwia czytanie dużych dokumentów dowolnego rozmiaru. (ap)

Komputery PC w samochodach

Intel Corporation w listopadzie tego roku przedstawił wizję wykorzystania mocy obliczeniowej oraz funkcjonalności komputerów osobistych w pojazdach mechanicznych. Intel chce współpracować z czołowymi firmami branży motoryzacyjnej, komputerowej, elektroniki użytkowej oraz telekomunikacyjnej nad opracowaniem platformy komputerowej, opar-

tej na procesorze Pentium. Uniwersalna platforma samochodowego systemu komputerowego łączy w sobie możliwości przetwarzania danych platformy PC z innymi technologiami takimi, jak telefonia komórkowa, komunikacja bezprzewodowa, Global Positioning System (GPS) i Radio Data Broadcast (RDB). (ap)

SAS w intranecie ratuje ekologię

Agencja Ochrony Środowiska USA (Environmental Protection Agency - EPA) za pośrednictwem WWW śledzi i kontroluje zanieczyszczenia środowiska naturalnego i niebezpieczne odpady produkcyjne na ziemi, w wodzie i w powietrzu dzięki rozbudowanej aplikacji do analizy statystycznej. EPA zatrudnia w dwunastu ośrodkach na terenie kraju 20 000 pracowników. Obecnie jest instalowane oprogramowanie, które zapewnia uproszczony, szybki dostęp do przeszło 50 największych baz danych środowiskowych i do współpracującego z WWW programu analizy statystycznej SAS na różnorakich platformach komputerowych. Jednym z atutów EPA są ogólnokrajowe bazy danych, dotyczących substancji zanieczyszczających środowisko naturalne, przyjętych zasad postępowania, pomiarów jakości powietrza atmosferycznego, zarejestrowanych emisji zanieczyszczeń i udzielonych zezwoleń. Bazami zawierającymi dane o zanieczyszczeniach środowiska naturalnego zarządza oprogramowanie produkcji firm: Software AG, IBM, Lotus i Oracle. (ap)

Łączność transatlantycka

Alcatel zawarł kontrakt na dostawę transatlantyckiego światłowodowego systemu telekomunikacyjnego. System, w skład którego wchodzą dwa kable transatlantyckie, połączy Londyn z Nowym Jorkiem. Realizacja kontraktu ma zostać zakończona w 1998 roku. Połączenie będzie wykorzystywać najbardziej zaawansowane technologicznie wyposażenie transmisyjne SDH, co umożliwi transmisję z prędkością do 10 Gigabajtów na sekundę. (ap)

SAS Institute wspiera JavaStation firmy Sun

Na pokazie Java Enterprise Solutions w Nowym Jorku SAS Institute

Najnowszy model telefonu MicroTAC

MicroTAC International 8700 jest najnowszym modelem telefonu dla sieci GSM, oferowanym w Polsce przez firmę Motorola. Za jego pośrednictwem można rozmawiać bez przerwy przez 10 godzin, przy użyciu baterii niklo-wodorkowej 1600 mAh o zwiększonej pojemności. Jednocześnie czas działania w stanie czuwania wynosi do 170 godzin. MicroTAC 8700 zapewnia zaawansowane możliwości przesyłania danych - jest jednym z najszybszych telefonów na rynku. Wykorzystanie telefonu, po podłączeniu do komputera PC, umożliwia transmisję danych, poczty elektronicznej i faksów z prędkością do 9600 bps. Technologia kompresji danych firmy Motorola - Digital Data Fast umożliwia jeszcze szybsze transmisje z prędkością do 36 000 bps. Model ten również optymalizuje wykorzystanie powierzchni wyświetlacza - dobiera wielkość liter tak, aby były one zawsze najbardziej czytelne, co jest szczególnie istotne przy wykorzystywaniu aparatu w samochodzie. (ap)

Wireless Computing Initiative

Czołowi producenci przenośnego sprzętu komputerowego i telekomunikacyjnego ogłosili zawiązanie inicjatywy, dążącej do zwiększania w Europie możliwości łączności bezprzewodowej i wykorzystania jej w biznesie do przesyłania danych komputerowych. Członkowie Mobil Data Intiative pomogą europejskim przedsiębiorstwom używać sieci GSM do bezprzewodowego łączenia przenośnych komputerów PC z firmowymi sieciami, bazami danych czy Internetem. Mobil Data Initiative została utworzona między innymi przez firmy Intel, Ericsson, Compaq, IBM, Toshiba oraz Cellnet, Mannesmann-Mobilfunk, T-Mobil, Telia i Vodafone. (ap)

Karty kredytowe American Express i IBM

American Express wspólnie z IBM zamierzają wprowadzić system, który umożliwiłby często podróżującym biznesmenom szybszą obsługę na lotniskach, przy czym korzystaliby z pierwszej w świecie wielofunkcyjnej karty kredytowej, podróżując bez biletu. W grudniu obie firmy rozpoczną próby systemu American Express Corporate Card. Karty systemu będą testowane za pomocą unowocześnionych czytników kart w bramkach linii lotniczych American Airlines, zainstalowanych na 21 amerykańskich lotniskach.

Metoda elektronicznych kart kredytowych uwalnia podróżujących od wszelkich kłopotów związanych z pilnowaniem kart pokładowych na terenie lotniska. Podróżny przechodząc przez bramkę wkłada kartę do czytnika, uzyskuje potwierdzenie rezerwacji fotela w samolocie i może rozpocząć podróż.

American Express początkowo udostępni karty wybranym grupom własnych pracowników i IBM, mając nadzieję, że zaakceptuje ten pomysł IATA, międzynarodowa organizacja przewoźników lotniczych. (ap)

Microsoft® Publisher 97

Microsoft® Publisher 97 jest ulepszoną wersją popularnego programu małej poligrafii dla firm, działającego w systemie operacyjnym Windows®. Obecnie program został wyposażony w różnorodne narzędzia, ułatwiające małym firmom szybkie i łatwe tworzenie profesjonalnych publikacji takich, jak broszury, ulotki i biuletyny. Nowa wersja rozszerza możliwości wydawnicze o publikowanie elektroniczne, umożliwiając użytkownikom tworzenie stron WWW w tak łatwy sposób, jak zwykłych dokumentów drukowanych. (ap)

SAS Institute wspiera JavaStation firmy Sun

Na pokazie Java Enterprise Solutions w Nowym Jorku SAS Institute zademonstrował nowo utworzone aplety Javy. "Nowa JavaSation Sun'a i SAS'owe aplety Javy stwarzają swoim użytkownikom okazję do wykorzystania najnowocześniejszych rozwiązań hurtowni danych na tanich stacjach klientów sieciowych. JavaSation eliminuje koszty administrowania stanowiskiem użytkownika, natomiast aplety SAS'a umożliwiają zachowanie poziomu inteligencji oprogramowania" - powiedział Chip Kelly, kierownik programu WWW SAS Institute.

Zademonstrowane aplety udostępniają użytkownikowi interfejs wskaż-i-kliknij podczas wykonywania takich zadań, jak przeglądanie i zarządzanie danymi z serwera, sporządzenie podsumowań i generowanie grafiki. Aplet po uruchomieniu przenosi się na stację klienta i umożliwia dzielenie przetwarzania pomiędzy klienta i serwer. Pozwala to na centralne zarządzanie aplikacjami i rozmieszczanie ich w sieci, a także umożliwia Systemowi SAS korzystanie z zasobów klienta.

SAS Institute jest jedną z pierwszych firm, które dokonały wdrożenia czystej specyfikacji Javy dla APIJDBC, oferując otwarte i kompletne rozwiązania hurtowni danych i systemów decyzyjnych. (ap)

Twarde dyski Toshiby

Na polskim rynku pojawiły się najnowsze modele 2,5-calowych dysków twardych firmy Toshiba: MK3303MAN o pojemności 3,3 GB oraz MK2103MAV - bardzo cienki dysk o pojemności 2,16 GB. Pierwszy z nich ma wysokość 19 mm i wagę 220 g; charakteryzuje się on buforem pamięci podręcznej 128 KB i średnim czasem dostępu 13 ms. Model MK2103MAV przy wysokości 12,7 mm waży 155 g. Przy komunikacji za pośrednictwem ATA-3 pozwala na transfer danych rzędu 16,6 MB/s. 128 KB pamięci podręcznej umożliwia osiągnięcie średniego czasu dostępu na poziomie 13 ms. Dysk można kupić za pośrednictwem INTEL-Serwis, a jego cena wynosi ok. 2300 PLN+VAT. (ap)

Dwunaste Jesienne Spotkanie PTI - Mrągowo'96

W dniach 25-29 listopada odbyło się w Mrągowie już dwunaste forum, zorganizowane przez Polskie Towarzystwo Informatyczne. Spotkanie było adresowane głównie do osób odpowiedzialnych za decyzje dotyczące strategii rozwoju informatyki w przedsiębiorstwach. Konferencja stanowiła kontynuację sześciu Szkół Jesiennych PTI, odbywających się w Mrągowie w latach 1984-1989 oraz pięciu Forów Technologii Informatycznych (1991-1995).

Gospodarze spotkania zapewnili uczestnikom udział w wykładach, prezentacjach i panelach. Wykłady, zaprezentowane na konferencji, miały na celu przedstawienie aktualnego stanu badań prowadzonych na polskich uczelniach. Referentami byli m.in. profesorowie: Wojciech Cellary, Jan Madey, Leszek Pacholski oraz Krzysztof Zieliński.

W cyklu "Panele" zostały przewidziane m.in. panele: o ochronie danych osobistych, spotkanie na temat: "Za co płaci podatnik w informatyce?", z udziałem przedstawicieli: KBN, URM, MPiH, MEN oraz Ministerstwa Ochrony Środowiska. (ap)

Rozbudowa SIMPLE-BUSINESS

SIMPLE-BUSINESS został stworzony już przed sześciu laty i w tym czasie został zainstalowany u ponad 1500 firm na terenie całej Polski. Przede wszystkim przeznaczony jest do wspomagania działalności gospodarczej małych i średnich firm handlowych (prowadzenie księgowości, realizacja rozliczeń w różnych walutach, tworzenia list płac pracowników wraz ze sporządzeniem odpowiednich deklaracji, obsługi magazynów, sprzedaży, prowadzenia rejestrów zakupów i sprzedaży, tworzenia deklaracji VAT).

Do najważniejszych zmian zastosowanych w wersji 6.0 należą nowe funkcje, wprowadzone w module Obrót Magazynowy, jak np. możliwość dokonywania rozchodu z magazynu wg metody FIFO (First Input - First Output) oraz LIFO (Last Input - First Output). SIMPLE-BUSINESS v. 6.0 zapewnia zapis dat na czterech cyfrach. (ap)

Procesor Digitala znowu najszybszy

Procesor StrongARM SA-110 Digitala uzyskał najwyższy notowany wynik w testach CaffeineMark 2.01, stosowanych do badania wydajności terminali sieciowych. Wynik testu CaffeineMark umożliwia ocenę wydajności interpretacji apletów pisanych w języku Java. Rekord uzyskano na prototypie terminala sieciowego Digitala zbudowanym na bazie procesora StrongARM SA-110, zarządzanym systemem operacyjnym JavaOS.

Terminal Digitala okazał się w tych testach dwukrotnie lepszy niż produkowane przez Silicon Graphics i Sun Microsystems. *"Wyniki testu pokazują, że nasz procesor interpretuje aplety napisane w języku Java niezwykle szybko. Co więcej - oznacza to, że procesor jest doskonale zaprojektowany do realizacji tanich aplikacji, wykorzystujących pamięci o małej pojemności",* powiedział Leo Joseph, szef marketingu do spraw aplikacji internetowych w oddziale Digitala produkującym układy półprzewodnikowe.

Digital produkuje procesory serii StrongARM, taktowane zegarem od 100 MHz (115 MIPS) do 233 MHz (270 MIPS) na licencji Advanced RISC Machines Ltd. Procesory te są zasilane napięciem 1,65 V i kosztują od 29$ do 49$ przy sprzedaży w partiach po 10 000 sztuk. (js)

OptimusNet

Prezes Optimusa S.A. Roman Kluska 13 listopada 1996 oficjalnie powiadomił o utworzeniu polskich zasobów internetowych. Krakowski Oddział Optimus S.A. prowadzi Polskie Zasoby Internetu, które opierają się o największy internetowy katalog, sporządzony w języku polskim, dostępny dla wszystkich użytkowników na serwerze OptimusNet pod adresem: **www.onet.pl**. Z jego pomocą można się już łączyć z ponad 2500 miejscami w sieci, w tym z kilkunastoma światowymi przeglądarkami, serwisami informacyjnymi, serwerami FTP, listami dyskusyjnymi, IRC i innymi. Oprócz katalogowania stron dostępnych już w sieci, Optimus S.A. położył nacisk na opracowywanie informacji z takich dziedzin , których dotychczas w polskim

Internecie brakowało lub zostały opracowane niedostatecznie. Jest nadzieja, że komercyjne potraktowanie "tematu" Internetu przez Optimus S.A. spowoduje, że od tej chwili Polska nie będzie postrzegana jako pustynia informacyjna. (js)

Porozumienie Microsoft z Telekomunikacją Polską S.A.

13 listopada 1996, dyrektor Microsoft Sp. z o.o. Waldemar Sielski oraz Krzysztof Trzewik, dyrektor Centrum Systemów Teleinformatycznych w imieniu Telekomunikacji Polskiej S.A. podpisali porozumienie o współpracy w promowaniu ogólnoświatowej sieci Internet oraz usług i narzędzi internetowych, oferowanych przez Microsoft i TP S.A. W ramach współpracy Microsoft będzie m. in. przekazywać TP S.A. wszelkie informacje dotyczące narzędzi dla Internetu, udostępni ich wersje testowe oraz będzie informować o planach dotyczących produktów internetowych Microsoft.

Celem wspólnych działań jest podniesienie jakości i komfortu korzystania z zasobów sieci Internet oraz upowszechnienie dostępu do niej. TP S.A. przyjęła przeglądarki internetowe Microsoft Internet Explorer 3.0 PL dla Windows 95 oraz Microsoft Internet Explorer 2.01 PL dla Windows 3.x jako standard i będzie je rozprowadzała nieodpłatnie wśród swoich klientów korzystających z usług internetowych. Warto podkreślić, że przeglądarki Microsoft są jedynymi przeglądarkami w polskiej wersji językowej. (js)

Internet na dysku

Firma INITEL-Serwis, oficjalny dystrybutor firmy ELEKTROSON wprowadziła na rynek oprogramowanie WebGrabber służące do zapisywania informacji uzyskanych z Internetu na dysku CD. Aby móc korzystać z nowej technologii zapisu niezbędny jest Netscape Navigator™ 3.0 (dostarczany wraz z pakietem WebGrabber), dzięki któremu przesyłane pliki są bezpośrednio zapisywane na dysku CD z pominięciem dotychczasowej procedury. Program jest przystosowany do pracy pod Windows 3.1 oraz Windows 95. Przewidywana cena zestawu (kontroler i oprogramowanie) wyniesie ok. 1000 USD. (ak)

CD-RW

Na początku 1997 roku PHILIPS Elektronics zaprezentuje opracowaną przez siebie nową technologię wielokrotnego zapisu danych na krążkach CD. Compact Disc - ReWritable działają w podobny sposób jak tradycyjny dysk twardy; jest możliwa ingerencja w dane już zapisane. Jest to szczególnie przydatne w pracy z dużymi aplikacjami i systemami archiwizacji danych. CD-RW będą czytane przez obecnie używane napędy, jak również napędy DVD. (ak)

Digital obniża ceny

7 listopada 1996 - Digital znacznie obniżył ceny 64-bitowych komputerów rodziny AlphaServer, które osiągnęły bardzo dobre wyniki w testach TPC-C. Pauline Nist, wiceprezydent Digitala ds. serwerów tak komentuje ostatnią obniżkę *"Digital jako jedyny rozwija własną, pełną - zawierającą rozwiązania sprzętowe, systemowe i aplikacyjne - ofertę takich systemów. Właśnie teraz postanowiliśmy znacznie uatrakcyjnić ceny naszych komputerów rodziny AlphaServer".*

Digital znacznie obniżył ceny systemów AlphaServer 4000 i 4100. Jest to wynikiem spadku cen na układy pamięci RAM. Przykładowo, cena systemu AlphaServer 4100 5/400 z Windows NT, która obecnie w wersji podstawowej wynosi 38040$ została obniżona o 25%. Natomiast serwer AlphaServer 4000 5/300, który teraz kosztuje 30040$ potaniał o 23%. Ceny wszystkich modułów pamięci dla serwerów 4000, 4100 i 2000 zostały obniżone o 29%. (js)

W dniu 15 listopada 1996 roku zmarł tragicznie

Ś.P. ***Andrzej Górbiel***

dziennikarz i informatyk. Jego odejście jest ogromną stratą dla całego środowiska informatycznego.
Redakcja "Informatyki" składa wyrazy głębokiego żalu Jego rodzinie i kolegom.

MS Camcorder

Program MS Camcorder, wyprodukowany przez Microsoft, służy do rejestrowania czynności wykonywanych w środowisku Windows 95. Camcorder ma stanowić integralny element przygotowywanego pakietu Microsoft Office 97 i będzie zarazem odpowiednikiem Lotus ScreenCam.

Program jest dostępny bezpłatnie w Internecie pod adresem: http://www.microsoft.com/office/office97/camcorder. W archiwum zajmuje on nieco ponad 600 KB. Po uruchomieniu wystarczy wybrać funkcję rejestrowania i wykonywać wszystkie kolejne czynności. Zatrzymanie nagrywania pozwala zapisać plik w postaci sekwencji wideo (AVI) lub jako samodzielny plik EXE, który można odtworzyć bez udziału jakiegokolwiek programu. Camcorder pozwala również nagrywać dźwięk przez mikrofon. Program będzie m.in. przydatny do tworzenia obszerniejszych prezentacji udostępnianych na dyskach CD-ROM. (ap)

CodeBase 6.1

Firma Sequiter Software Inc. stworzyła kolejną wersję biblioteki do zarządzania bazami danych xBASE dla języków C, C++, Visual Basic i Delphi - CodeBase 6.1, Nowa wersja została wzbogacona o język Java. CodeBase obsługuje dane w formatach dBASE, Clipper, foxPro, Visual dBASE i Visual FoxPro wraz z indeksami i plikami memo. Programista może wybrać nie tylko język programowania, ale także system operacyjny i konfigurację. CodeBase jest jednym z najszybszych mechanizmów obsługi baz danych. (ap)

Cyfrowa metoda kolorowego druku

Firma Lexmark i Polaroid zdecydowały się na współpracę w celu konstruowania wysokojakościowych drukarek i ich dodatkowego wyposażenia. Przy produkcji zostanie wykorzystana metoda sześciobarwnych druków Lexmarka. *"Firma Polaroid jest przekonana, że sześciobarwny druk jest kluczowym elementem umożliwiającym drukowanie realistycznych zdjęć na tanich drukarkach"* - powiedziała Margaret Wolf z firmy Polaroid. (ap)

Zaprosili nas...

Informix na warsztaty prasowe dotyczące tworzenia infrastruktury informacyjnej dla Internetu/intranetu na przykładzie OnLine Workgroup Server.

Zarząd Targów Warszawskich na konferencję prasową poświęconą VI Międzynarodowym Targom Telekomunikacji KOMTEL, Konferencji EUROINFO oraz wystawie INTERNET-EXPO.

Wydawnictwo Naukowe PWN na promocję "Komputerowego słownika języka polskiego PWN".

MacroSoft na konferencję prasową, dotyczącą dotychczasowej działalności firmy i nowej strategii jej rozwoju oraz na uroczyste obchody jubileuszu 10-lecia istnienia firmy.

Publishing Institute, Linotype-Hell & Zakład Poligraficzny Unigraf na "Dni otwarte" z okazji uruchomienia pierwszej w Polsce pełnoformatowej naświetlarki firmy Linotype-Hell "Signaseiter" oraz na seminaria dotyczące nowoczesnych technologii poligraficznych.

Polskie Centrum Badań i Certyfikacji na konferencję prasową, w ramach Europejskiego Tygodnia Jakości oraz na Wystawę Jakość'96.

OPTIMUS SA na konferencję prasową oraz prezentację nowego kierunku działalności firmy.

IBM Polska i Soft-tronik Polska na konferencję prasową, w której uczestniczyli Mirosław Szturmowicz, dyrektor generalny IBM oraz Stanisław Winiarski, prezes Soft-tronik Polska.

OPTIMUS SA na konferencję prasową z okazji podpisania umowy o współpracy handlowej pomiędzy Disney Interactive i OPTIMUS SA.

SUN Microsystem na promocyjną imprezę firmy pn. "Java EnterpriseSolutions".

IDG Poland S.A. na Premierę Wirtualnego ComNetu, pierwszego działania w ramach Wirtualnego Rynku oraz ogłoszenie programu ComNet Warszawa '97.

Borland i BSC Polska na konferencję prasową z okazji powstania oficjalnego ogólnopolskiego wsparcia technicznego dla produktów firmy Borland.

Intergraph Europe Polska na prezentację nowej wersji oprogramowania SOLID EDGE v2 po polsku (ap).

Konkurs "Polska w Internecie"

W listopadowej i zarazem pierwszej edycji konkursu "Polska w Internecie", Jury pod honorowym przewodnictwem Ryszarda Horowitza w składzie: Marzena Donajski, Juliusz Donajski, Henryk Gajewski, Marta van der Haagen, Jerzy Sychut, Wojciech Sylwestrzak postanowiło przyznać Nagrodę Miesiąca stronie: DELPHI SUPER PAGE (http://sunsite.icm.edu.pl/~robert/delphi/).

Strony te istnieją od lipca 1995 roku i są prowadzone przez Roberta M. Czerwińskiego. Treść ich jest adresowana do pasjonatów programowania w języku delphi i na tej płaszczyźnie strona znajduje stałe grono swoich odbiorców na całym świecie. Dowodem tego może być chociażby kilkanaście kopii stron "Delphi" na pięciu kontynentach. Autor strony tworzy forum wymiany informacji dla użytkowników oprogramowania i umożliwia wymianę doświadczeń oraz aktualizację oprogramowania. Forma prezentacji zaproponowana przez autora jest przejrzysta, co ułatwia łatwe i szybkie odnajdywanie interesujących informacji.

Następny werdykty Jury i następne wyróżnienia zostaną ogłoszone 1-go stycznia. Dokładne informacje na temat konkursu można przeczytać na stronie WWW: http://psi.org.pl/konkurs/ oraz http://www.kbn.gov.pl/konkurs/

Zgłoszenia do kolejnych edycji konkursu przyjmowane są pod adresem: Maga@ikp.atm.com.pl.

Marzenna Donajski - Sekretarz Jury

Normowanie systemów F/K

W Kołobrzegu 23-25 września w ramach konferencji INFRA'96 spotkali się księgowi i informatycy, aby dyskutować nt.: "Normowania systemów rachunkowości wspomaganej komputerem". Gośćmi konferencji byli wykładowcy z Paryża, którzy przedstawili referaty na temat normowania we Francji. Tematyka konferencji była pochodną prac wspólnego zespołu PTI/SK, pracującego od pewnego czasu nad opracowaniem normy zawodowej tej klasy systemów. (Na mocy decyzji Prezesa PTI i Prezesa SK powołano zespół wspólny 5 księgowych i 5 informatyków dla opracowania założeń do normy zawodowej systemów F/K).

Dyskusja była bardzo twórcza i w sposób znaczący ukierunkowała dalsze prace Komisji. Konferencje zorganizował prof. Dziedziczak - przewodniczący Komisji wspólnej, a stronę organizacyjną zagwarantował Oddział Szczeciński Stowarzyszenia Księgowych.

Istotnym ustaleniem konferencji było to, że komisja wspólna PTI/SK przygotuje pierwszą wersję normy zawodowej dla systemów informatycznych F/K. Wersja ta zostanie zaprezentowana na specjalnej konferencji zorganizowanej na wiosnę przyszłego roku przez Polskie Towarzystwo Informatyczne. Celem konferencji będzie dalsze zbieranie opinii środowiska księgowych i informatyków tak, aby kolejna wersja miała szanse stanowić wzorzec dla opracowywanych i rozprowadzanych systemów klasy F/K. Jest to pierwszy krok w długim procesie zmierzającym do certyfikacji systemów informatycznych.

Zainteresowanych tematyką prosimy o kontakt z prof. Dziedziczakiem lub z Biurem PTI - Leszkiem Bogusławskim Sekretarzem Generalnym (jerome@warman.com.pl).

Zdzisław Szyjewski

Biuletyn PTI

Zarząd Główny Polskiego Towarzystwa Informatycznego na posiedzeniu 9 listopada br. ustalił, że biuletyn PTI będzie drukowany w postaci wkładki w miesięczniku "Informatyka". ZG PTI powołał panią dr inż. Ewę Łukasik na redaktora naczelnego tego biuletynu. Ma się on ukazywać w kolejnych numerach "Informatyki", począwszy od stycznia 1997 r. (lw)

VI Walne Zgromadzenie Polskiej Izby Informatyki i Telekomunikacji

10 stycznia 1997 r. odbędzie się konferencja nt. "Dokonania i rola Polskiej Izby Informatyki i Telekomunikacji w kształtowaniu polskiego rynku teleinformatycznego" z udziałem zaproszonych gości ze środowiska informatycznego. Po konferencji odbędzie się VI Walne Zgromadzenie PIIiT, zwołane na podstawie Uchwały Rady Izby z dnia 21 listopada 1996 r. (ap)

Piractwo w Internecie

Dzięki wielomiesięcznemu dochodzeniu, prowadzonemu przez grupę Novell'a do zwalczania piractwa komputerowego, we wrześniu br. został ujęty 27-letni technik komputerowy, nazywający sam siebie "Piratem". Udostępniał on każdej osobie, mającej połączenie z Internetem, nielicencjonowane produkty Novell'a o wartości 60 000 USD, a także oprogramowanie komercyjne produkcji innych członków BSA (Business Software Alliance). Podczas przeszukiwania inspektorzy Novell'a znaleźli również pliki zawierające informacje dotyczące sposobów wykonywania bomb i instrukcje umożliwiające nielegalne korzystanie z kart kredytowych. "Piratowi" grozi kara więzienia do 3 lat i grzywna do 10 000 franków szwajcarskich. (ap)

Fabryka multimediów

W wiedeńskiej siedzibie Alcatel'a Telecom powstało centrum interaktywnych aplikacji multimedialnych. Oferuje ono potencjalnym użytkownikom szerokopasmowych sieci ATM możliwość "jazdy testowej" po autostradzie informacyjnej. Centrum jest wyposażone w platformę, umożliwiającą uruchomienie aplikacji nowej generacji, tj. przekazu wideo oraz superszybkiego transferu danych. Platforma została zaprojektowana przez Alcatel Telecom, Oracle, nCube oraz Sequent.

"Fabryka" świadczy takie usługi, jak np.: projekt i konsultacje, zapewnienie infrastruktury (serwery, symulacja sieciowa, środowisko dla rozwoju, testowanie, itd.), stworzenie potrzebnych aplikacji, kodowanie i testowanie zawartości, system operacyjny, transfer know-how dla klienta. Billa, największa sieć sprzedaży detalicznej w Austrii, połączyła 50 swoich sklepów poprzez "fabrykę multimediów", aby zaoferować swoim klientom telezakupy. (ap)

32-bitowa karta Ethernetu

Xircom zaprezentował pierwszą 32-bitową kartę Ethernetu wykonaną w postaci karty PCMCIA w nowym standardzie CardBus. CardBus Ethernet Adapter 10/100 współpracuje z komputerami przenośnymi, wyposażonymi w nowy interfejs CardBus. Jest to interfejs, który naśladuje w komputerach typu notebook architekturę PCI stosowaną w komputerach biurkowych. Umożliwia on przesyłanie danych bez przerywania pracy procesora, dzięki odpowiedniemu sterowaniu magistralą. Transfer odbywa się po 32-bitowej magistrali taktowanej zegarem o częstotliwości 33 MHz. Pierwszymi notebookami, wyposażonymi w interfejs CardBus są komputery przenośne Toshiby serii Tecra oraz NEC'a - Versa 6000.

Karta CardBus Ethernet Adapter współpracuje ze standardem Ethernet oraz Fast Ethernet, automatycznie rozpoznając prędkość transmisji i dostosowując się do niej. Dysponuje też agentem SNMP i jest wyposażona w programy-sterowniki do wszystkich popularnych systemów sieciowych.

Karta - zasilana napięciem 3,3 V - kosztuje 255 dolarów (1410 USD w pakietach po 6 sztuk, zaś 4600 - w zestawach handlowych po 20 sztuk). (ap)

Ośrodek intranetowy Compaq Inline

Firma Compaq Computer Corp. ogłosiła światową premierę Compaq Inline, ośrodka intranetowego z zasobami dla własnych pracowników. Dostęp do Compaq Inline można uzyskać posługując się dowolna przeglądarką WWW. Ośrodek ten umożliwia pracownikom firmy bezpieczną komunikację, przesyłanie danych oraz przetwarzanie transakcji.

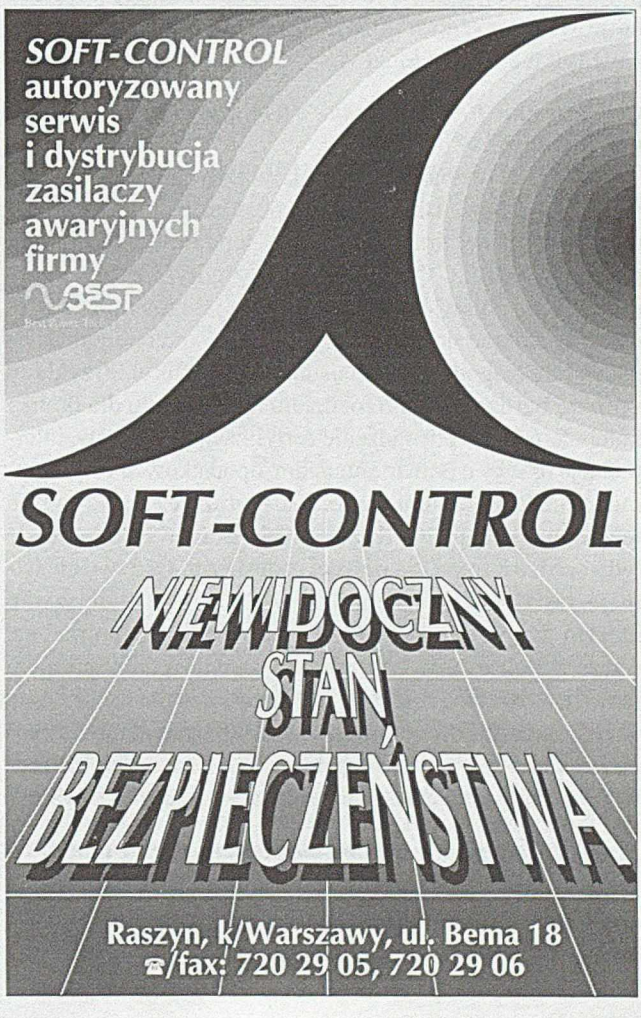
Internet Compaq'a działa w oparciu o serwery Compaq ProLiant (najlepsze serwery wg opinii Computer Reseller News). Sieci intranet obejmują aplikacje działajace na wielu różnych platformach sprzętowych. (ap)

SMP w OS/2 Warp

IBM wprowadził funkcję symetrycznego przetwarzania wieloprocesorowego (SMP) do systemu OS/2 Warp Server Advanced. Umożliwi to znaczne zwiększenie wydajności przetwarzania i połączenie serwerów sieciowych, co oznacza w praktyce niższe koszty eksploatacji i uproszczone adminstrowanie siecią. Testy, przeprowadzone przez IBM, wykazały, że OS/2 WarpServer Advanced z SMP osiąga o ponad 25% lepsze wskaźniki wydajności niż Microsoft Windows NT Server 3.51.

OS/2 WarpServer Advanced z obsługą SMP jest 200 do 300 procent szybszy niż OS/2 2.11 SMP w przypadku funkcji intensywnie wykorzystujących procesor, jak np. aplikacje baz danych czy tworzenie aplikacji.

Stosowanie SMP przez dotychczasowych uzytkowników OS/2 WarpServer Advanced będzie możliwe przed końcem bieżącego miesiąca w ramach bezpłatnego uaktualnienia. (ap)



Izba firm teleinformatycznych

Z dr. Wacławem Iszkowskim prezesem Polskiej Izby Informatyki i Telekomunikacji rozmawia Lesław Wawrzonek

Z czego wynika potrzeba istnienia Izby?

Gdyby stosunki gospodarcze w Państwie były jasno i precyzyjnie ustalone, Izba nie byłaby potrzebna - tak jednak nie jest. Ciągłe zmiany zasad prowadzenia działalności gospodarczej powodują, że wciąż pojawiają się ustalenia niedopasowane do siebie. Ponadto regulacje prawne często nie uwzględniają specyfiki branży teleinformatycznej. Dlatego właśnie musi istnieć reprezentacja grupy firm zajmujących się podobną działalnością, np. w formie izby gospodarczej, w imieniu których prezentuje zastrzeżenia co do niektórych regulacji prawnych Parlamentu oraz administracji. Reprezentanci firm wnoszą też do nich propozycje zmian lub odpowiednich interpretacji. W takich przypadkach istotne jest, aby reprezentanci firm teleinformatycznych najpierw uzgodnili pomiędzy sobą wspólne stanowisko, uwzględniając różne interesy, a dopiero potem było ono prezentowane władzom administracyjnym. Z tych powodów Izba jest potrzebna i co warto zauważyć, z jej działalności wynikają korzyści dla całego środowiska firm teleinformatycznych, a nie tylko dla jej członków.

Czy wszystkie firmy związane z teleinformatyką powinny należeć do Izby?

Znacząca liczba firm, członków Izby, na pewno widzi taką konieczność. Bardzo często pożytek z przynależności do Izby dla danej firmy ujawnia się dopiero w sytuacji, gdy powstaje konflikt. Oczywiście nikt z góry nie wie, kiedy może on się pojawić. Istniejący potencjał Izby można wtedy wykorzystać do podjęcia skutecznej interwencji polegającej na złagodzeniu lub usunięciu konfliktu. Przykłady - proszę bardzo. Sposób rozdziału kontyngentu dla branży komputerowej, wprowadzenie certyfikacji produktów, interpretacja ustawy o prawie autorskim, opodatkowanie oprogramowania, zamówienia publiczne oraz regulacja rynku telekomunikacyjnego. Każda z tych spraw w swoim czasie była bardzo istotna dla pewnej grupy firm. Oczywiście istnieje jeszcze wiele firm, szczególnie tych mniejszych, którym przynależność do Izby nie wydaje się potrzebna. Część z nich uważa, że i tak inni rozwiążą dany problem bez ich udziału. Jestem przekonany, że z czasem coraz więcej zarządów firm dojdzie jednak do wniosku, że wspólne działanie jest bardziej efektywne niż samodzielne rozwiązywanie problemów, które dotyczą większej liczby firm.

Po co Izba zajmuje się promowaniem rynku teleinformatycznego?

Dla Izby jest to działalność nieco marginesowa, ale niezwykle istotna, gdyż przynosi pożytek wszystkim firmom oraz poszerza zakres oddziaływania Izby. Ważne jest też, że dla klientów firm teleinformatycznych Izba staje się pewnym gwarantem rzetelności firm do niej należących. Już bardzo często zdarzają się klienci, którzy poprzez Izbę dowiadują się o firmach mogących dostarczyć im dobre oferty. Również dla organizacji, firm i instytucji zagranicznych Izba jest często pierwszym kontaktem w Polsce, przez który zaczynają poszukiwać partnerów do współpracy lub inwestycji. Izba wielokrotnie już pomagała nawiązać takie kontakty.

Co jest według Pana aktualnie najważniejszym zadaniem Izby?

Oprócz zajmowania się problemami działalności gospodarczej w branży teleinformatycznej, aktualnie najważniejsze dla członków PIIiT jest powołanie zespołu rzeczoznawców. Powinni oni brać udział w rozstrzyganiu coraz bardziej złożonych merytorycznie konfliktów pomiędzy dostawcami, a ich mniej zadowolonymi z usług i dostaw klientami. Rzeczoznawcy mogliby być także wykorzystywani przez klientów do przygotowywania skomplikowanych zapytań ofertowych oraz do nadzoru realizacji kontraktów. Moim zdaniem powstanie zespołu rzeczoznawców jest pierwszym krokiem do weryfikacji merytorycznej projektów i wykonania systemów informatycznych przez uprawnionych do tego specjalistów (wzorem budownictwa, instalacji elektrycznych i wielu innych dziedzin).

Jest Pan inicjatorem współpracy Izby z innymi organizacjami oraz stowarzyszeniami teleinformatycznymi - w jakim celu?

Chcemy konsolidować środowisko teleinformatyczne i stworzyć warunki do stałej wymiany opinii pomiędzy różnymi grupami interesów. Dlatego też Izba współpracuje ze wszystkimi organizacjami. Wspólną platformą do dyskusji wszystkich tych organizacji może być Raport z I Kongresu Informatyki - „Strategia Rozwoju Informatyki w Polsce". Na jego podstawie możemy dochodzić do nowych ustaleń i proponować wspólnie przyszłe działania. Wszyscy bowiem - i firmy i ich klienci oraz zespoły informatyczne - jedziemy na tym samym wózku. Wózek ten się jeszcze czasem mocno kolebie, ale stałoby się źle, gdybyśmy z niego pospadali. Środowisko teleinformatyczne nie może funkcjonować w rozbiciu. Oczywiście musi być spełniony jeden warunek - musimy wzajemnie szanować swoje poglądy i odrębność. Walki polityczne pozostawmy partiom i frakcjom politycznym - informatyka jest jedna, bez względu na to, która strona będzie rządzić.

Dziękuję za rozmowę.

Rozważnie, ale i szybko - czyli rzecz o tym, jak wybierać system informatyczny wspomagający MRPII

Aleksander Popończyk

Powszechnie wiadomo, że system informatyczny trzeba wybierać rozważnie. Tak nakazuje zdrowy rozsądek oraz fakt, że jest to jedna z tzw. inwestycji strategicznych - wymagających dodatkowego, głębokiego zastanowienia i przemyśleń. Skutkiem tego zwraca się mniejszą uwagę na czas trwania procesu wyboru. Umyka przy tym fakt, że przy wyborze systemu wspomagającego MRPII właśnie czynnik czasowy powienien być postrzegany jako najważniejszy. Dla średniej wielkości przedsiębiorstwa każdy miesiąc bez MRPII to utrata znacznych korzyści, wg APICS'u - rzędu 100 tys - 300 tys dolarów amerykańskich.

Zabobony wyboru systemu

Jak wspominałem w poprzednich artykułach, system informatyczny jest elementem niezbędnym do funkcjonowania rozbudowanego systemu MRPII w nowoczesnym przedsiębiorstwie. On po prostu musi być. Obecnie nie jest problemem znalezienie systemu zgodnego ze standardem MRPII. Co więcej - systemów jest całe mnóstwo tak, że nie wiadomo który wybrać. W sytuacji, gdy mamy wybór, staramy się w sposób mniej czy bardziej racjonalny ocenić dostępne warianty, aby wybrać rozwiązanie „najlepsze", cokolwiek by to określenie oznaczało. Ponieważ w skali kraju zakupiono już ponad setkę różnych rozwiązań tej klasy, możemy mówić, że przyjęła się pewna praktyka prowadzenia procesu wyboru systemu. Niektóre z jej założeń są słuszne i sensowne, aliści są przemieszane z teoriami wręcz niebezpiecznymi - zabobonami. O niektórych z nich powiem poniżej parę słów.

Należy analizować wszystkie opcje. W koncepcji tej pobrzmiewa echo idealizmu, że gdy się rozważy każdy dostępny system, to stwierdzi się, który z nich jest najlepszy. Po pierwsze - nie zdarza się, aby dwie osoby uważały to samo i tak samo za najlepsze. Wybór jest zatem z natury rzeczy czymś subiektywnym, czemu dopiero próbuje się nadać znamiona obiektywizmu przez przeróżne wartościowania, skale i oceny. Oceny cząstkowe poszczególnych funkcji badanego systemu są nadawane bardzo arbitralnie, zaś uzyskaną ocenę zbiorczą traktuje się jak jakąś prawdę objawioną, która wskazuje nam nieomylnie ów „najlepszy" system. Niejeden system kupiono na bazie uzasadnienia, że np. suma jego ocen jest o 2 punkty większa niż konkurenta drugiego w kolejności.

Oprócz trudności z obiektywizmem wyboru należy również brać po uwagę czas, jaki pochłania analiza kilkunastu ofert różnych systemów. Średnia oferta ma około 150 stron, nie licząc załączników i dokumentacji systemu. Ktoś to musi przynajmniej raz przeczytać, zrozumieć i ocenić. Aby praca ta nosiła znamiona rzetelnej, to zajmuje ona - wraz z ewentualnymi wizytami referencyjnymi oraz wyjaśnieniami - około 2 tygodni na każdą propozycję. Rozpatrywanie 6-8 umów to jest już kwartał. Jeżeli, średnio, pełny system informatyczny, włączając oprogramowanie, wdrożenie, kable i sprzęt kosztuje w granicach 600 tys - 1 mln USD, a oferty wahają się w granicach +/- 30%, to więcej stracimy deliberując miesiącami o zaletach i wadach poszczególnych systemów, niż zyskamy na wyborze oferty „najlepszej" z jakichś względów. Globalnie, za pieniądze uzyskane dzięki szybkiemu wdrożeniu MRPII w przedsiębiorstwie, nawet względnie słabe rozwiązanie przystosujemy do naszych potrzeb, bowiem każdy miesiąc bez MRPII to strata 100 tys - 300 tys dolarów amerykańskich (za APICS'em).

Kupujemy system informatyczny i od razu nadaje się on do użytku. Nadzieja, że zainstalowane oprogramowanie będzie działać w sposób zgodny z oczekiwaniami, jest co najmniej złudna. Każde oprogramowanie, bez względu na to, co mówią sprzedawcy, posiada błędy. Jedne systemy mniej, a drugie więcej, ale zawsze znajdzie się coś wymagającego poprawy. Dodatkowo każdy nabywca ma swoją unikalną specyfikę - może przetwarzać informacje w sposób niespotykany gdzie indziej lub może rejestrować pewne dane, których projektanci systemu nie przewidzieli. Nie zdarza się, aby system od razu spełniał wszystkie potrzeby - coś trzeba będzie na pewno zmienić. A znając naszą rzeczywistość, łatwiej jest zmienić system niż nawyki organizacji. Należy także wziąść pod uwagę, że instalacja nowego systemu nie odbywa się w próżni; zwykle z jakichś systemów informatycznych już korzystamy. Niezbędne będą zatem wszelkiej maści interfejsy i programy do konwersji danych. Lepiej sobie od razu powiedzieć, że będziemy modyfikować oprogramowanie i starać się panować na zakresem rozszerzeń, ich kosztami i czasem.

Wystarczy zakupić system informatyczny, aby MRPII zaczęło funkcjonować. Mam nadzieję, że sofistyczny charakter tego stwierdzenia jest już oczywisty dla Czytelnika. To ludzie przyczyniają się do poprawnego funkcjonowania systemu. Nie wolno zatem rozpatrywać tych zagadnień w oderwaniu od siebie. Równolegle z wyborem systemu trzeba kontrolować proces przyswajania pryncypiów MRPII przez pracowników . System informatyczny będzie efektywny dopiero, gdy na stałe zagoszczą w przedsiębiorstwie takie cechy, jak odpowiedzialność i sumienność oraz

szerokie zrozumienie funkcji i roli MRPII. Nie należy także zapominać o danych. Muszą one być dokładne i kompletne. Jeżeli nie jesteśmy w stanie tego warunku spełnić, to po co kupować już teraz system informatyczny. Do takiego zakupu musimy być przygotowani organizacyjnie, a nie tylko informatycznie.

Im system bogatszy w różne funkcje, tym lepiej. Mówi się, że od przybytku głowa nie boli. Oj boli, boli! Zakładając, że wiemy, czego chcemy, powinniśmy szukać dokładnie takiego systemu, który spełnia potrzeby, i nie oglądać się na inne. Wcale nie musi to być ten wyposażony w największą ilość funkcji. Jeżeli zaś nie wiemy tak do końca, czego potrzebujemy, to też powinniśmy skoncentrować się na czymś ograniczonym. W przeciwnym wypadku zostaniemy zalani lawiną opcji i możliwości, przekraczającą nasze zdolności analizy. W prostej linii prowadzi to do wielomiesięcznych wdrożeń, polegających na badaniu kolejnych funkcji systemu i zastanawianiu się, czy ich użyć, a jeśli już, to do czego oraz w jaki sposób. Dla biznesu przedsiębiorstwa koszt takiego eksperymentu jest horrendalny.

Najnowsza wersja systemu jest najlepsza. W pełni zrozumiała jest chęć kupowania jak najnowszych modeli różnych rzeczy. Na nieszczęście, na grunt informatyki nie należy przenosić wszystkich doświadczeń z życia codziennego. Tutaj bardziej liczy się stabilność i niezawodność, niż dwa nowe przyciski na ekranie interfejsu graficznego. Nowy produkt najprawdopodobniej będzie posiadał nowe błędy. Borykanie się z nimi jest ceną, jaką płacimy za podążanie za nowinkami. Można powiedzieć, że zwykle wersja systemu o pełnym numerze, tj 1.0, 2.0, 3.0 itd, nie nadaje się do użytku. Trochę lepiej jest, gdy pojawiają się poprawki do tych wersji, oznaczane zwyczajowo literami a,b,c, etc. Należy wówczas szukać jak „najwyższej" litery. Zwykle wersje, np. 2.1 czy 3.2, są już dopracowane. Z autopsji wiem, że firmy „zachodnie" często decydują się świadomie na starszą wersję oprogramowania, wiedząc, że będą miały z nią mniej problemów natury technicznej.

Standardowe umowy są nienegocjowalne. Jeżeli decydujemy się na zakup systemu, który wzbudził nasze zainteresowanie, to zawieramy umowę kupna, zwaną często kontraktem. Jest to moment emocjonujący i często umyka uwadze, że jest to najbardziej newralgiczna chwila wyboru. Tutaj możemy stracić wszystko za jednym zamachem. Najczęściej dzieje się tak, gdy damy się omamić stwierdzeniem dostawcy, że ma on standardowy kontrakt, który wszyscy podpisują. Proszę się siebie zapytać, dlaczego my nie możemy być potraktowani w sposób wyjątkowy? Wszak to my mamy pieniądze. Jeżeli dostawca ma sztywne stanowisko w tej kwestii, to czemu przypuszczamy, że przy wdrożeniu będzie się nam lepiej współpracować i nie będziemy lekceważeni? Każdy kontrakt można negocjować, a przynajmniej zawrzeć z dostawcą czy reprezentantem dostawcy, aneks do standardowego kontraktu.

Kontrakt zawiera się z tylko z jednym partnerem - integratorem. Od blisko 4 lat pewną manierą jest zawieranie umów na dostawę, instalację i wdrożenie z jednym, głównym partnerem. Oferty składają konsorcja, kierowane przez lidera, ponoszącego główną odpowiedzialność kontraktową za prace. Sądzi się bowiem, że taka struktura zapewni szybsze rozwiązywanie konfliktów między firmami dostarczającymi różne elementy składowe systemu, usprawni zarządanie wdrożeniem oraz nie wymaga od kupującego dysponowania kadrą znającą wszechstronnie różne zagadnienia techniczno-organizacyjne wdrożenia. Nic bardziej błędnego. Dziadowi już mówili, że pańskie oko konia tuczy. Jeżeli sami nie będziemy wszystkiego pilnować, to trwać będziemy w złudnym przekonaniu prawidłowej współpracy aż do nieprzyjemnego ocknięcia się, gdy coś nie będzie działać. Dlaczego zakładamy, że integrator będzie zatrudniał jakąś osobę znającą całość zagadnień, jeżeli my sami nie chcemy takiego „omnibusa" zaangażować? Najprawdopodobniej zadanie „integrowania" zostanie zlecone osobie dysponującej wolnym czasem, a nie specjalną wiedzą merytoryczną. Nie łudźmy się również, że echa wewnętrznych konfliktów między stronami konsorcjum nie będą do nas docierać. Na pewno będziemy ubierani w rolę arbitra albo opóźnienia będą wyjaśniane obiektywnymi kłopotami poszczególnych dostawców. Dzieje się tak, bo tak naprawdę to tylko my jesteśmy zainteresowani tym, żeby system zadziałał, i to możliwie szybko. Dostawca od momentu podpisania kontraktu ma nas w ręku i jest prawie niemożliwe wygrać z nim w sądzie czy w arbitrażu. Warto pamiętać, że integrator za sam fakt występowania w takiej roli pobiera zwykle ok. 10% wartości całego kontraktu. Są to duże pieniądze i jeżeli zaproponujemy je wybranej przez nas, doświadczonej osobie, jako wynagrodzenie w ramach umowy o dzieło, to bądźmy pewni - będzie sprawy pilnowała jak własnej, a na pewno lepiej będzie dbała o interes zakładu, niż jakikolwiek integrator. Jeżeli nie mamy własnego pracownika, którego zadaniem jest **kontrolowanie różnych dostawców**, to na pewno znajdziemy kogoś na rynku lokalnym, np. w firmie komputerowej za bramą. Nie bójmy się zawierać umów z różnymi stronami na dostawę sprzętu, oprogramowania czy wdrożenia. Rozsądnie podchodząc do rzeczy podołamy koordynacji prac.

Zdrowe zasady wyboru

Aby nie być posądzonym o czarnowidztwo, pragnę stwierdzić, że na szczęście praktyka wyboru systemów dostarcza również pozytywnych wzorów. Poniżej opisuję kilka zdroworozsądkowych kroków, wynikających z mojego doświadczenia. Jeżeli ktoś byłby zainteresowany całościowym omówieniem procedury wyboru systemu informatycznego wspomagającego MRPII, to polecam publikację p. Christophera D. Gray, pt. „The right choice", wydaną przez nieocenioną oficynę Oliver Wight.

Edukacja. Generalnie każde dobrze rokujące podejście rozpoczyna się od edukacji. Zarówno kadra kierownicza, jak i zespół wybierający oprogramowanie muszą dysponować przynajmniej podstawowymi wiadomościami o przedmiocie zakupu. Jest to niezbędne do przeprowadzenia rzeczowej oceny i świadomego wyboru. Edukacja, zwykle w formie seminarium lub szkolenia prowadzonego w ramach wstępnej fazy wdrażania systemu MRPII, powinna pozwolić pracownikom na ukształowanie wizji roz-

wiązania. Ludzie nie tylko muszą wiedzieć, co to jest MRPII, ale również jak będzie ono mogło wyglądać w ich konkretnych warunkach.

Komisja selekcyjna. Wybór systemu jest zwykle wykonywany kolegialnie. Można takiej formie zarzucać zachowawczość, ale wydaje się ona być najbezpieczniejsza. Celem wydajnej pracy komisja musi zostać właściwie upełnomocniona do prowadzenia prac. Ludzie wtedy traktują rzecz serio, gdy wiąże się z nią odpowiedzialność bądź autorytet. Skład komisji powinien być adekwatny do jej zadania, co oznacza, że musi być w niej m.in. szef produkcji, główny technolog oraz informatyk. Komisja musi wiedzieć, jaką kwotą dysponuje na zakup systemu.

Jasno określone wymagania. Jeżeli nie ma jasno określonego celu, każda droga do niego jest jednakowo zła. Przed rozpoczęciem całego procesu, a najpóźniej w jego wczesnych fazach, konieczne jest zdefiniowanie wymagań funkcjonalnych odnośnie nabywanego systemu. Mogą to być listy wymagań, może to być wybór ze wspomianej już normy przemysłowej „Standard system", ewentualnie każdy inny dokument stwierdzający wystarczająco jasno, jednoznacznie i szczegółowo, czego poszukujemy. Wymagania przekazujemy oferentowi do wglądu, a następnie, w trakcie specjalnego spotkania staramy się uzyskać jednoznaczne odpowiedzi. Dokument, zawierający wymagania i opis ich spełnienia przez system, jest integralną częścią oferty oraz późniejszego kontraktu.

Źródła oprogramowania. Już na wstępie należy rozważyć różne dostępne źródła oprogramowania. Członkowie komisji powinni zdecydować o tym, czy pakiet będzie budowany na zamówienie, tworzony własnymi siłami czy też będzie to gotowe rozwiązanie. Jeżeli przekracza to ich kompetencje, mogą sięgnąć po fachowe porady, np. konsultantów zewnętrznych. Jednak zawsze powinna zapaść ostateczna decyzja odnośnie konkretnego źródła dostawy systemu. Jeżeli będziemy mieszać ze sobą różne koncepcje, to nie mamy szans na sprowadzenie ich do wspólnego mianownika.

Znaleźć pakiet. Jeżeli decydujemy się na zakup gotowego rozwiązania, to wiedząc już czego i gdzie szukamy, pozostaje nam zidentyfikować potencjalnie interesujące rozwiązania. Otwarty przetarg jest tutaj ostatecznością, zajmuje bowiem dużo czasu, a jego efekty są nad wyraz mizerne. Raczej należy wybrać 2-3 „polecanych" dostawców i poprosić ich o ofertę. Musimy pamiętać, że na szybko przeprowadzonej analizie zyskamy więcej, niż ewentualnie „utracimy" na wyborze oferty o kilka procent droższej od innych.

Ocena ofert i dokumentacji. Dokumenty, które przekażą nam oferenci, należy wnikliwie przeczytać. Uwaga! - oferta piękna i wymuskana wcale nie musi być bardziej wartościowa od innych. Zwykle oznacza to tylko, że firma ma dobrze przygotowany wzór oferty, w który - we właściwe miejsca - wprowadza nazwę naszej firmy. Jeżeli zaś oferta jest nieskładna, strony są nie po kolei oraz czegoś w niej brakuje, to znaczy, że była kończona w przeddzień złożenia i nikt jej nie czytał. Nie jest to coś dyskwalifikującego, ale powinno być ostrzeżeniem, że firma może być nadmiernie obciążona lub panuje w niej bałagan. Wybór dokumentacji stanowi zwykle obowiązkowy załącznik do oferty. Ponieważ firmy raczej nie mają w zwyczaju dołączać najgorszych jej fragmentów, to możemy być prawie pewni, że nic lepszego ponad to już nie otrzymamy. Warto poświęcić trochę czasu na jej uważne przestudiowanie.

Ocena wymagań sprzętowych i zaawansowania technicznego rozwiązania. Zwykle dostawca proszony jest o podanie w ofercie zalecanej, względnie minimalnej konfiguracji sprzętu komputerowego. Propozycję tę musimy porównać z posiadanym wyposażaniem i trendami rynku. Trzeba też ocenić poziom techniczny, jaki rozwiązanie sobą reprezentuje. Czy jest skalowalne, czy są dostępne kody źródłowe i możliwy jest samodzielny rozwój aplikacji? Zasadniczo nie akceptuje się już rozwiązań np. typu TurboPascal v5.5 w oparciu o Betrieve i Novell Network.

Czy system działa? Sprawa wydaje się prozaiczna, ale trzeba bezwzględnie zobaczyć system w działaniu. Dokładnie ta wersja, która jest proponowana, powinna być przedmiotem kilkudniowych badań, np. w siedzibie oferenta. Można zawsze poprosić konsultanta o asystę i o wykonanie jakichś funkcji na systemie. Zobaczymy od razu, czy naprawdę możliwe jest to, co czytamy w broszurach reklamowych; jak skomplikowane jest uzyskanie tych informacji oraz, co też jest istotne, jaki poziom merytoryczny prezentują sobą wdrożeniowcy firmy.

Określenie niezbędnych modyfikacji i interfejsów. Jeżeli jakieś rozwiązanie wydaje nam się interesujące, należy ocenić, ile czasu zajmą i ile będą kosztować poprawki i rozszerzenia zidentyfikowane w wyniku porównania możliwości systemu z wymaganiami. Kto będzie je wykonywał? Czy programista jest osiągalny i czy dysponuje odpowiednią wiedzą? Co z prawami autorskimi i gwarancjami? Czy opracowane programy można będzie przenieść do nowych wersji systemu czy też trzeba je będzie pisać od nowa? Komisja musi uzyskać jasne odpowiedzi na wszystkie wątpliwe kwestie.

Oszacowanie kosztów. W kolejnym kroku musimy uczciwie określić koszty zakupu rozwiąznia. To znaczy, ile gotówki i na kiedy będziemy musieli wyłożyć w perspektywie trwania wdrożenia. Również należy oszacować nakłady wewnętrzne, będące pochodnymi szkoleń i testów. Nasi pracownicy, oddelegowani do prac nad systemem, również nas kosztują. Kierownictwo zawsze żąda tych informacji, zatem bądźmy na to przygotowani. Starajmy się tylko nie przywiązywać do kosztów systemu zbyt dużej wagi. Nie powinien być to jedyny warunek przesądzający o odrzuceniu czyjejś oferty. Wszystko jest kwestią umowy, a margines cenowy, w jakim porusza się oferent, jest naprawdę duży.

Wizyty referencyjne. Nic tak nie oddaje zalet i wad systemu, jak rozmowa z jego użytkownikami. Telefonicznie możemy uzyskać pewne informacje, ale również powinniśmy się wybrać i zobaczyć, jak inni wdrażają bądź wdrożyli rozwiązanie w rzeczywistości. Przestrzegam jednak przed bezkrytycznym przyjmowaniem opinii zebranych w trakcie wizyt. Jak wspominaliśmy, z różnych powodów większość wdrożeń idzie raczej nie najlepiej, i dlatego też

trzeba patrzeć na to zdroworozsądkowo, próbując dociec prawdziwych przyczyn porażek. Sądzę, że przeważnie przyczyną jest brak wiedzy odnośnie organizacyjnych aspektów podejścia do wdrożenia MRPII. Pochodną tego stanu rzeczy jest wątpliwa korzyść z wizyt u krajowych użytkowników. Na pewno nie są one szkodliwe, ale co nam po informacji, że system u kogoś nie działa. W tym przypadku zalecam raczej wizyty referencyjne za granicą i poczynienie obserwacji, co to znaczy, że ktoś pracuje w systemie MRPII posługując się danym oprogramowaniem. Będziemy mieli wtedy piękny wzór do naśladowania.

Pozycja finansowa dostawcy i dostępne wsparcie. Sprawą o dużym znaczeniu jest również stabilność finansowa oferenta. Można zauważyć występowanie korelacji pomiędzy zasobami finansowymi firmy z jej potencjałem wdrożeniowym. Oczywiście małe firemki, np. początkujący dystrybutorzy, nie powinny być dyskryminowane, ale należy się im bacznie przyjrzeć, czy przypadkiem nie są to obrotni sprzedawcy bez wsparcia zespołu merytorycznego. Już wiemy, że od dostawcy nie będziemy wymagali ogromnych prac wdrożeniowych - to jest działka samego przedsiębiorstwa - ale ktoś musi z sensem wyjaśnić, jak funkcjonuje oprogramowanie, jak się je parametryzuje czy wreszcie ktoś musi poprawiać błędy.

Partnerska umowa

Najtrudniejszą oraz - co się zdarza - najbardziej czasochłonną fazą wyboru systemu są negocjacje kontraktowe i podpisanie umowy. Jest to okres dużych emocji i wymaga od obydwu stron silnej woli i zdecydowania. Niestety, często nabywca jest w trakcie negocjacji na słabszej pozycji. Po pierwsze oferent widzi już siebie w roli wybranego dostawcy - wszak nie rozmawialibyśmy z nim tylko, aby sobie pogawędzić - i usztywnia swoje stanowisko. Zawodowy sprzedawca czuje się tej roli jak ryba w wodzie. Potrafi się dziwić, obrażać, a nawet pokrzykiwać. Wszystkie chwyty są tutaj dozwolone. Po drugie nabywca ma zrozumiałe skrupuły przed „konfrontacją". Myśli perspektywicznie o współpracy i obawia się, czy jego upór w pewnych kwestiach nie będzie odbijał się negatywnie na późniejszych stosunkach. Po trzecie na ogół nie ma doświadczenia i „na wiarę" przyjmuje zapewnienia oferenta o zabezpieczeniu jego interesów.

Uwaga - to wszystko jest tylko grą pozorów. Zwykle sprzedawca jest osobą otrzymująca prowizję od sprzedaży. Zatem obiecałby on i złote góry, byleby tylko kontrakt został podpisany. Nie bójmy się z nim twardo negocjować. Możemy być prawie pewni, że po zawarciu kontraktu nie zobaczymy więcej już tego sprzedawcy w naszym przedsiębiorstwie. Pamiętajmy jednak o tym, że negocjowana umowa musi być partnerska, tzn. akceptowana przez obydwie strony i zabezpieczająca ich interesy. Są tylko dwie możliwości - albo wygrywamy razem albo traci przedsiębiorstwo kupujące system. Dlaczego? Jeżeli kontrakt jest niekorzystny dla dostawcy, to on się z niego i tak nie wywiąże, gromadząc zawczasu „papiery" na rozprawę lub likwidując interes. Jeżeli umowa jest niekorzystna dla zakładu, to oferent wykorzysta każdą możliwość, aby ograniczyć swój wkład w prace. W obydwu wariantach tej możliwości straci na tym przedsiębiorstwo.

Przystępując do negocjacji musimy być przygotowani. Komisja powinna zapoznać się szczegółowo z ofertą, przedyskutować refleksje po wizytach referencyjnych, prezentacjach, etc. W spotkaniach muszą brać udział kompetentni ludzie, i to po obydwu stronach. Aby rozmowa miała sens, mają być oni uprawnieni do podejmowania wiążących rozstrzygnięć. Niezbędny jest również radca prawny, najlepiej znający problematykę umów na dostawę systemów informatycznych. Ciekawą rzeczą jest, że korzystne jest dla przedsiębiorstwa wynajęcie konsultantów zewnętrznych do prowadzenia negocjacji. Jest to pewnego rodzaju trzecia strona, dająca zespołowi czas na zastanowienie się w trakcie rozmów oraz podnosząca kwestie niejasne bądź sporne. Często doradcy dysponują przykładami ustaleń z innych kontraktów, co pozwala przełamywać impasy.

Zasadniczą rzeczą pozostaje, że to, czego nie ma w kontrakcie, nie zostało ustalone. Dlatego też trzeba włączyć do kontraktu nasze wymagania i odpowiedzi udzielone na nie przez dostawcę. Wszelkie zidentyfikowane modyfikacje, kwestie praw autorskich i sprawy gwarancji. Również nie wolno zapomnieć o karach umownych, choć nie spotkałem się jeszcze z przypadkiem, aby udało się je wyegzekwować. Koniecznie trzeba opisać testy i procedury testowania systemu. Od ich pomyślnego wykonania zależeć będzie, czy odbierzemy system czy nie. Określić należy również obsługi poinstalacyjne i pogwarancyjne, oraz tzw. maintenance - opiekę nad systemem.

Samowystarczalność

Nadchodzi wreszcie podniosła chwila, gdy wszystkie ustalenia są dopięte na przysłowiowy ostatni guzik. Podpis, uścisk dłoni, lampka szampana i życzenia owocnej współpracy. Jest to znak, że już trzeba myśleć, jak się rozstać z dostawcą. Bynajmniej nie jest to makabryczny makiawelizm, ale proste następstwo faktu, że dostawca zajmuje się sprzedażą, a my rzeczywistym użytkowaniem oprogramowania. Im wcześniej będziemy w stanie absolutnie samodzielnie i niezależnie obsługiwać program, tym lepiej. Musimy dopracować się własnej kadry do tworzenia modyfikacji, raportów i obsługi operatorskiej. Jeżeli funkcjonowanie całego naszego biznesu będzie opierać się o ten system, to czy pozwolimy sobie na ryzyko jakiejkolwiek zależności od zewnętrznej firmy? Nie - musimy być tak przygotowani, jakby dostawca mógł z dnia na dzień przestać istnieć. Wtedy nas nic nie zaskoczy.

Mam nadzieję, że ten artykuł będzie służył praktyczną pomocą osobom odpowiedzialnym w przedsiębiorstwach produkcyjnych za wybór systemu informatycznego klasy MRPII, a przynajmniej da pewien asumpt do własnych przemyśleń. Jeżeli omawiana tematyka Państwa interesuje, zapraszam za miesiąc, gdzie w kolejnym artykule powiemy sobie o metodach wdrażania systemu MRPII.

Aleksander Popończyk jest pracownikiem polskiego oddziału firmy Coopers & Lybrand, zajmującej się doradztwem w zakresie zarządzania.

Jak budować system Computer Integrated Manufacturing?

TECHNOLOGIA

Józef B. Lewoc, Antoni Sydor

Artykuł niniejszy oraz drugi tych samych autorów („Jak budować system CIM? - - Realizacja") są oparte na doświadczeniach własnych i tezach opublikowanych na kilku prestiżowych konferencjach (m.in.: L.K. Błach, K. Cybińska-Piróg, J.B. Lewoc, M. Rozent, West-East approaches of automation, A case study, CAA'91 (IFAC), Krems 1991; J.B. Lewoc, T. Ciukszo, P. Pawelec, An economic way to CIMs, CIM'92 (IFAC), Wiedeń 1992; J.B. Lewoc, L.Z. Misiura, A. Ruszkowska, A low cost DS application in a big power plant, LCA (IFAC), Wiedeń 1992.) Nie były one dotąd publikowane w Polsce, gdyż w latach silnej recesji tematyka ta mogłaby być uznana za subkategorię systems science fiction.

Jednak obecnie, przy odbudowie polskiej gospodarki, staje się ona coraz bardziej istotna i powstają realne możliwości implementacji, co uzasadnia zgłoszenie artykułów do druku.

Od informatyka zamiast wstępu

Czytałem kiedyś wypowiedź pewnego samodzielnego pracownika naukowego, który w oficjalnej recenzji wyraził się mniej więcej tak: „Computer Integrated Manufacturing (CIM) to fabryki przyszłości, hale pełne robotów, taka Japonia XXI wieku". Ten pogląd może przywołać na myśl bojowy okrzyk rycerzy gwiezdnych, skutecznie kiedyś wprowadzony do literatury przez Stanisława Lema. Jeśli nawet pominąć tu fakt, że ponad dwa miliony Polaków nie posiada dziś stabilnego zajęcia, to jak zrozumieć paradoks, gdy zastępca dyrektora generalnego centrum projektowania systemów komputerowych i telekomunikacyjnych największego producenta takich robotów w Japonii zastanawia się poważnie, co mi odpowiedzieć na żartobliwe pytanie: „Tylko dlaczego one nie chcą nic kupować?", a naukowcy w naszym Kraju snują przed nami takie wizje CIM. I chcą je rozwijać, aby zapewnić nam świetlaną przyszłość. Może warto byłoby postawić postulat, by przesunąć im wypłaty wynagrodzeń na tę właśnie przyszłość?

Oba artykuły zostały napisane wspólnie przez specjalistę od technicznego przygotowania i kontroli produkcji oraz przez informatyka. Należy wyraźnie podkreślić, że choć można być dumnym z tego, co w swoim zawodzie osiągają informatycy, to przecież nie mają oni szans, by konkurować z ludźmi, którzy reprezentują często wielowiekowe tradycje i dorobek nauki w zakresie samego wytwarzania. Wręcz przeciwnie, naprawdę duże osiągnięcia wdrożeniowe i teoretyczne uzyskują tylko ci, którzy dostatecznie wcześnie zrozumieją, że informatyka wnosi do wytwarzania jedynie szybkość, z jaką można wykonywać proste i często żmudne czynności. Nową jakość w wytwarzaniu można uzyskać jedynie wtedy, gdy systemy informatyczne budowane są wespół ze specjalistami w dziedzinie przygotowania i prowadzenia produkcji, organizacji itd.

Bazę dla niniejszych artykułów stanowią doświadczenia uzyskane przy projektowaniu rzeczywistych systemów CIM (w szczególności dla przykładowego przedsiębiorstwa przemysłu elektromaszynowego - Zakłady). Nie wymieniamy ich z nazwy, nie chcąc wprowadzać akcentów indywidualnych, lecz by przedstawić problemy występujące powszechnie.

Cel systemu CIM

Cele systemu CIM nie mogą być rozpatrywane same w sobie - należy je odnosić do celów działalności przedsiębiorstwa w warunkach obecnych i w przyszłości. Trzeba określić funkcję celu oraz zbiór ograniczeń, którym podlega przedsiębiorstwo, co jest warunkiem podejmowania poprawnych decyzji projektowych i inwestycyjnych dla realizacji systemu CIM.

Dla Zakładów zespół projektowy przyjął, że podstawowe ograniczenie tezy realizacji funkcji celu powinna stanowić jakość wyrobów, obejmująca pełny okres ich projektowania i życia - od badań rynkowych, przez koncepcję, założenia, projekt, konstrukcję, technologię, produkcję, zbyt, serwis, aż do eliminacji (likwidowania zużytych wyrobów). Jako miarę wymaganej jakości wyrobów przyjęto stan magazynów końcowych w odniesieniu do wielkości produkcji w okresie odpowiednim dla charakteru działalności Zakładów.

Za funkcję celu przyjęto minimum kosztów działalności Zakładów. Zatem przyjęto, że celem działalności Zakładów jest minimalizacja kosztów tej działalności, przy czym na koszty składają się wydatki przedprodukcyjne, produkcyjne i poprodukcyjne, a warunkiem, jaki ma być przy tym zachowany, jest odpowiednia jakość produkcji mierzona skuteczną sprzedażą. Jest to równoważne z maksymalizacją produktywności, czyli efektywnego wykorzystania wszystkich czynników produkcji (w szczególności środków pracy, materiałów czy nośników energetycznych) pod warunkiem utrzymania jakości wyrobów zapewniającej odpowiednią sprzedaż wyrobów.

Powyższe sformułowanie jest dość ogólne, aby można je było stosować do wielu przedsiębiorstw, a sposób realizacji funkcji celu musi być umiejscowiony w środowisku, do którego się odnosi. Dla Zakładów produkujących kilka rodzajów (asortymentów) i typów wyrobów postanowiono indywidualnie rozpatrywać poszczególne typy wyrobów (ponumerowane indeksem i) ze zbioru typów I. W tym przypadku funkcja celu została określona jako Minimum (koszty przedprodukcyjne$_i$ + koszty produkcyjne$_i$ + koszty poprodukcyjne$_i$, gdzie i $\in$ I) po wszystkich i ze zbioru I pod warunkiem, że roczny zbyt i-tego wyrobu nie jest mniejszy od jego produkcji rocznej (okres roczny przyjęto ze względu na sezonowość zapotrzebowania rynku).

Teraz zadanie systemu CIM staje się jasne: powinien ułatwić realizację i kontrolę spełnienia funkcji celu Zakładów oraz ograniczeń jakościowych. I tu powstaje pierwsza przesłanka dla architektury informacyjnej (programowej) systemu CIM: powinna wzajemnie jednoznacznie odwzorowywać strukturę systemu organizacyjnego przedsiębiorstwa. Konieczność izomorficznej relacji jest oczywista: tylko taka współzależność eliminuje kolizje pomiędzy obo-ma systemami.

Dla Zakładów przyjęto propozycję docelowego modelu organizacyjnego złożonego z Dyrektora Generalnego (GX), trzyosobowego zarządu odpowiedzialnego za przygotowanie produkcji, prowadzenie produkcji i zadania ogólne oraz dziewięcioosobowej dyrekcji kierującej wykonywaniem zadań branżowych: DQ - zarządzanie jakością, DR - działalność badawczo-rozwojowa, DT - techniczne przygotowanie produkcji, DP - zaopatrzenie materiałowe i ruch, DI - zapewnienie warunków ruchu, DE - księgowość i finanse, DM - marketing, DS - serwis i DN - zarządzanie ogólne.

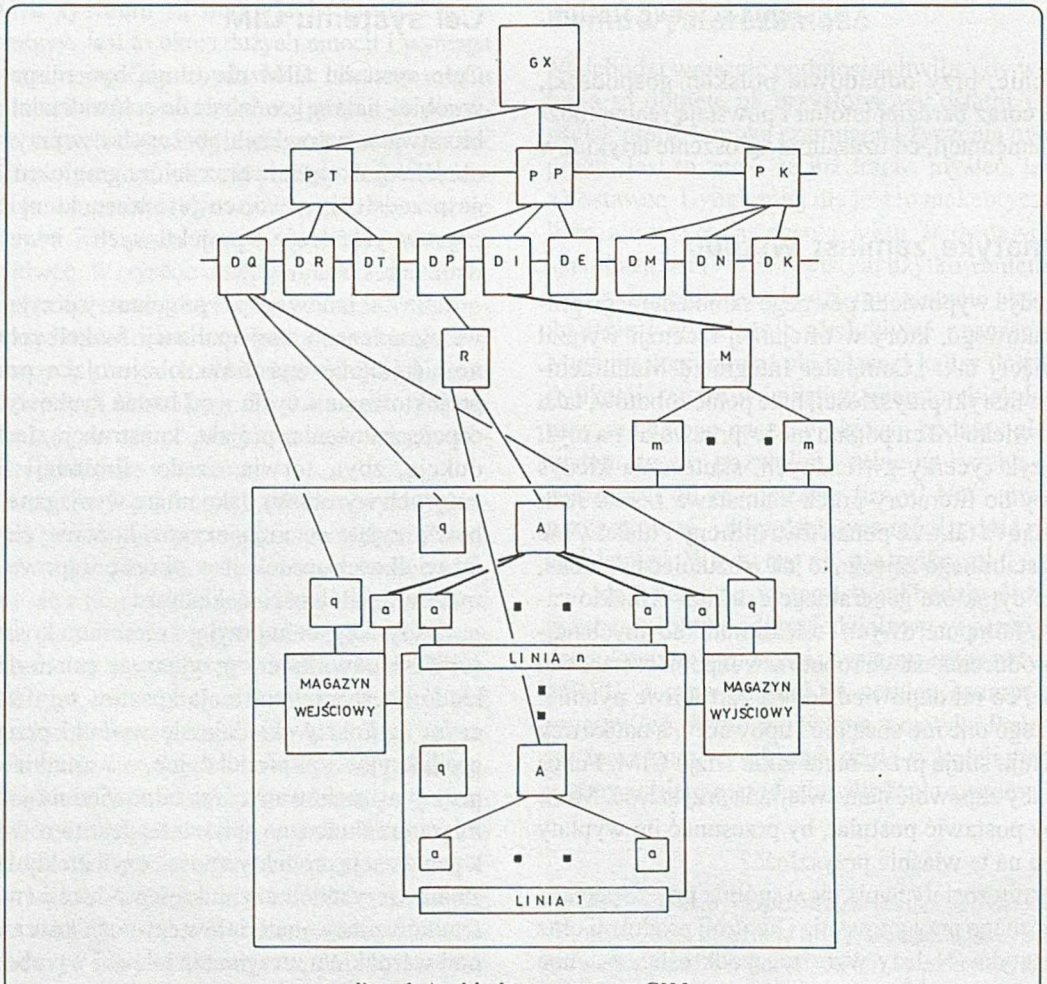
Architektura informacyjna systemu CIM

Gdy uwzględni się drzewiastą strukturę Zakładów i ich systemu produkcyjnego, to architektura informatyczna systemu CIM powinna mieć postać pokazaną na rysunku 1.

Zwracamy uwagę na odniesienie architektury informacyjnej CIM do systemu produkcyjnego Zakładów i na zbiór zintegrowanych, komputerowych podsystemów zarządzania i kontroli jakości (q), automatyki (R, A, a – przez to rozumiemy przede wszystkim komputerowe systemy monitoringu produkcji) oraz systemów kontroli nośników energii i sterowania źródłami/odbiorami (M, m – od systemu Media). Tego się na ogół dziś nie robi z uwagi na brak zintegrowanych systemów zarówno w Polsce, jak i na świecie (sytuacja najbardziej zbliżona do docelowej, jaką prezentujemy, jest najbardziej rozwinięta w energetyce).

Z punktu widzenia optymalnego prowadzenia przedsiębiorstwa, na architekturze z rys. 1 należy wdrożyć dwa podstawowe podsystemy:

- zarządzanie i kontrola jakości (Q-TEST),
- ocena kosztów (KOSZTY).



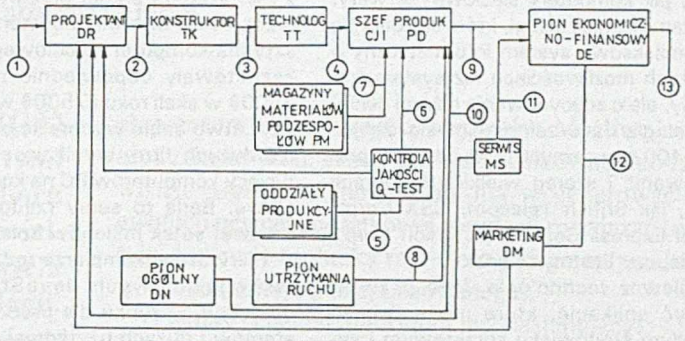
Rys. 1. Architektura systemu CIM

Q-TEST dla Zakładów pomyślano tak, aby docelowo spełniać wytyczne i wymagania norm międzynarodowych - np. ISO 8402 (terminologia) i ISO 9000-9004 (Zarządzanie jakością i systemy zapewniania jakości) czy ich odpowiedników WE (EN 29402, EN 29000-29004). Rozwiązanie szczegółowe zaproponowano tak, by każdy wyrób otrzymywał komputerowe świadectwo kontroli jakości obejmujące fazę przedprodukcyjną, produkcji i poprodukcyjną. Świadectwo to ma być dowodem zachowania poprawnych procedur wytwarzania (Good Manufacturing Procedures) i pozwalać na sprawne wyszukiwanie odpowiedniej dokumentacji techniczno-produkcyjnej czy handlowej, umożliwiając odtworzenie historii ruchu wyrobu. Dla niektórych produktów wymaga to zastosowania nowoczesnych środków diagnostycznych (np. termowizja), a dla wszystkich - zintegrowanego systemu Q-TEST na bazie lokalnych podsystemów q z rys. 1.

Dla podsystemu KOSZTY opracowano model przepływowy, pokazany w uproszczeniu na rys. 2. Szacowanie kosztów rozpoczyna się od projektanta, który analizuje potrzeby, rynek, prognozy popytu, możliwości realizacji itp. (1) i tworzy koszty projektowe (2). Konstruktor opracowuje konstrukcje i koryguje koszty projektowe (2), wyznaczając koszty konstrukcyjne (3). Z kolei technolog, opracowując technologię, wyznacza koszty technologiczne (4), które stanowią bazę do obliczania kosztów produkcji (9). Jednak tu muszą być uwzględnione odchylenia od kosztów technologicznych (inaczej standardowych, normatywnych czy teoretycznych) na podstawie kosztów wytwarzania (5) skorygowanych przez kontrolę jakości do postaci wynikowych kosztów wytwarzania (6). Wprowadzane są też rzeczywiste koszty materiałów (7), które obejmują też części, podzespoły oraz koszty pomocy warsztatowej.

Dodatkowe źródło kosztów to piony utrzymania ruchu (DI) i ogólny (DN). Wprowadzane są koszty nośników energetycznych (8) (energia elektryczna, gaz, para, woda, ścieki itp.) a także innych czynników niezbędnych do utrzymania ruchu - ochrona zdrowia czy działalności szkoleniowo-kulturalnej (10). Należy zwrócić uwagę, że koszty te powinny być związane z indywidualnymi wyrobami, choć, oczywiście, np. koszty ochrony zdrowia należy dzielić według pewnego klucza pomiędzy wszystkie egzemplarze produktów finalnych.

Rzeczywiste koszty produkcji (9) uzupełnione o koszty serwisu (w tym gwarancyjnego) (11) a także badań marketingowych i reklamy (12) są wykorzystywane przez pion ekonomiczno-finansowy (DE) do obliczania kosztów Zakładów (13).



Rys. 2. Uproszczony model przepływowy systemu obliczania kosztów

Zwróćmy uwagę na fakt, że bez zintegrowanego systemu informatycznego taki model rachunku kosztów jest nierealizowalny. Zakłady produkują około miliona wyrobów rocznie, przy czym nawet co sto egzemplarzy może nastąpić zmiana lub odstępstwo od normatywnej technologii. Zatem należy zgromadzić i przetworzyć dane o około dziesięciu tysiącach różnych partii rocznie. Bez zintegrowanego wspomagania komputerowego wykonanie takiego zadania jest nierealne.

Stąd wymaganiem stawianym systemowi CIM jest zapewnienie możliwości pełnej i dokładnej analizy kosztów dzięki szybkości i niezawodności zbierania, przetwarzania i przechowywania danych w stopniu istotnie wyższym, niż możliwy do osiągnięcia w tradycyjnym systemie ręcznym (nawet wspomaganym komputerowo).

Sposób realizacji systemu CIM przedstawimy w kolejnym artykule.

Józef B. Lewoc jest pracownikiem BPB Leader (Leading Designer) we Wrocławiu

Antoni Sydor jest pracownikiem Przedsiębiorstwa Aparatury Spawalniczej we Wrocławiu

Digital w wielu wymiarach?

INFORMACJE

21 października b.r. Arbor Software Corporation, wiodący na świecie producent systemów klient/serwer do zastosowań biznesowych ogłosiła, że Essbase, wielowymiarowa baza danych będąca produktem tej firmy, będzie oferowana przez Digital na komputerach rodziny AlphaServer z systemem operacyjnym Windows NT. Wersja beta Essbase, działająca na maszynach Alpha z systemem NT została zaprezentowana przez obie firmy podczas dorocznej konferencji Dimensions, która odbyła się w dniach 21-23 października w Santa Clara.

Dla firmy Arbor jest to możliwość znacznego rozszerzenia bazy sprzętowej, na której działa baza Essbase. Serwery rodziny AlphaServer z systemem NT stanowią idealną platformę dla aplikacji, które opierają się o złożone przetwarzanie, na bieżąco wielkich masywów danych analitycznych OLAP (on-line analytical processing).

„Serwery rodziny AlphaServer z systemem NT są niezwykle istotną platformą systemową dla naszej bazy danych", powiedział James Dorrian, prezes i dyrektor generalny Arbor Software. Zestawienie serwerów Digitala i bazy Essbase, powoduje, że korporacje będą w sposób ekonomiczy stosować technologie OLAP. Trzeba mieć świadomość, że te technologie bazują na gigabajtach informacji krytycznych z punktu widzenia prowadzenia biznesu w ramach przedsiębiorstwa przez tysiące użytkowników". (j.sz)

Sun odkrywa karty

18 listopada 1996 Sun Polska zorganizował kolejne z cyklu światowych spotkań, którego celem było zaprezentowanie technologii Java przeznaczonej dla organizacji gospodarczych. Technologia Java obejmuje wiele elementów systemu sieciowego takich, jak komputery sieciowe, serwery, oprogramowanie i usługi, które składają się na kompleksowy system informatyczny o większych możliwościach niż system tradycyjny, ale o zdecydowanie niższej cenie. Technologią Java zainteresowało się już ponad 400 niezależnych producentów oprogramowania i szereg wielkich korporacji takich, jak British Telecom, CSX Corp., Federal Express Corp., First Union Corp., FTD Inc, czy Eastman Kodak.

Ponieważ technologia Java pozwala tworzyć aplikacje, które mogą działać w każdym środowisku sprzętowym i systemowym, naturalną konsekwencją jej zastosowania jest przemieszczanie oprogramowania i pamięci z terminali i stacji roboczych do sieci i serwerów. Ewidentną korzyścią płynącą z niezależności oprogramowania od platformy sprzętowej i systemowej jest, w przypadku firm posiadających sieci z tysiącem węzłów, natychmiastowa obniżka kosztów od 50 do 80%, jak szacują specjaliści, wynikająca z „odchudzenia" komputerów klienckich, które zostaną zastąpione komputerami sieciowymi. Oferta Sun Microsystems zawiera rodzinę komputerów sieciowych, JavaStation, która nie wymaga żadnego nadzorowania, pełną linię serwerów sieciowych Netra™j z systemem napisanym w technologii Java oraz kompleksowe oprogramowanie i usługi, które są przeznaczone dla prowadzenia i wspierania działalności gospodarczej.

Komputer sieciowy JavaStation w pełnej konfiguracji będzie kosztował mniej niż 1000$, natomiast konfiguracja bazowa około 750$. Jednakże główne oszczędności w stosunku do systemów typu PC będą wynikały z eksploatacji komputera sieciowego. Przeciętny koszt użytkowania systemu PC w ciągu roku jest szacowany na 11900$, a w ciągu trzech lat na 35700$, w tym koszty sprzętu i oprogramowania stanowią tylko około 21%. Według badań przeprowadzonych przez firmę Sun Microsystems takie koszty dla komputera sieciowego będą się kształtowały odpowiednio na poziomie 2500$ w skali roku i 7500$ w skali trzech lat. Łatwo sobie wyobrazić oszczędności dla dużych firm, wynikające z wymiany tysięcy komputerów PC na komputery sieciowe. Będą to sumy rzędu dziesiątek, a nawet setek milionów dolarów.

Pierwsze egzemplarze rodziny komputerów sieciowych JavaStation będą dostępne na rynku dla producentów systemów i dużych użytkowników pod koniec grudnia bieżącego roku. Produkcja ruszy pełną mocą w lutym 1997 roku. Komputer sieciowy będzie lekki i nie większy od typowego notebooka. Nie będzie posiadał twardych ani elastycznych dysków, żadnych gniazd rozszerzeń, ani napędu CD-ROM, ani też żadnych ruchomych elementów, o które trzeba będzie się martwić. Podstawowymi komponentami komputera sieciowego będą: układ microSPARC™II, pamięć o pojemności od 8 do 64 MB, port sieciowy wykonany w technologii 10BaseT, który w połowie roku 1997 będzie wykonywany w technologii 100BaseT oraz kolorowy monitor XVGA 14 lub 17 calowy.

Rozwiązania sprzętowe uzupełniają produkty programowe, które mają zapewniać prostotę użytkowania komputera sieciowego oraz odpowiedni interfejs typu przeglądarki, służącej do realizacji wszystkich zadań, poczynając od instalowania potrzebnego oprogramowania, aż do zarządzania heterogenicznym środowiskiem sieciowym. Są to przede wszystkim **Hot- Java™ Views, JavaOS™, Solaris™ Internet i Solstice™.** (js)

IFS APPLICATIONS w Soda-Mątwy

IZCh Soda Mątwy S.A. jest największą fabryką sody w Europie. Instalacja IFS obejmuje licencję na 40 użytkowników i dotyczy gospodarki remontowej, materiałowej i finansów.

Na początku b.r. uzgodniono zakres modyfikacji ułatwiających pracę z systemem. Obecnie trwa normalna eksploatacja wszystkich trzech systemów. W ramach systemu finansowego od lipca uruchomiono moduły: IFS/Księga Główna, IFS/Księga Należności i IFS/ Księga Zobowiązań, służące do sporządzania bilansu miesięcznego. Trwają prace nad uruchomieniem modułu IFS/Środki Trwałe. System IFS/Dystrybucja pracuje w całości. Główne modyfikacje to obsługa sprzedaży komisowej i tak zwane raporty dedykowane dla klienta. Trwają prace nad systemem informowania kierownictwa. Fabryka jest zainteresowana systemem MIS, opracowanym ostatnio w warszawskim biurze IFS. (js)

IFS w PTK Centertel

Najnowszym kontraktem IFS Poland jest umowa z firmą PTK Centertel - operatorem pierwszej w Polsce sieci telefonii komórkowej. Po okresie gwałtownej rozbudowy infrastruktury telekomunikacyjnej i zdobywaniu klientów, Centertel uznał, że tak rozbudowana sieć wymaga rozwiązań systemowych. Utrzymanie w ruchu i prawidłowa eksploatacja rosnącej liczby stacji przekaźnikowych stają się z dnia na dzień ważniejsze. Wybór IFS APPLICATIONS spośród konkurencyjnych ofert ICL, Oracle i SAP, był poprzedzony staranną analizą, którą wykonano w rekordowym czasie trzech tygodni. Obejmowała ona analizę księgowości i środków trwałych, zaopatrzenia i gospodarki magazynowej oraz służb utrzymania ruchu, a także analizę potrzeb odnośnie sprzętu komputerowego. (js)

Porozumienie MCI z Digitalem

W połowie października zostało zawarte porozumienie pomiędzy MCI Systemhouse a firmami Microsoft, Marcam i Digital, na mocy którego zostanie podjęta współpraca w zakresie projektowania i sprzedaży kompleksowych rozwiązań informatycznych dla przedsiębiorstw przemysłowych. Oferta będzie adresowana w szczególności do producentów działających w branżach chemicznej, spożywczej, farmaceutycznej, hutnictwa metali i szkła oraz papierniczej.

W ramach kompleksowych systemów, integrowanych z już działającymi u użytkowników aplikacjami, będzie oferowany sprzęt, oprogramowanie i usługi firm, które podpisały porozumienie. Taka koncepcja chroni istniejące inwestycje informatyczne, równocześnie dając klientom możliwość dalszego rozwijania własnych systemów. Firma Marcam zapewni podstawowe oprogramowanie typu ERP (Enterprise Resource Planning), wspomagające planowanie działalności gospodarczej przedsiębiorstwa, natomiast firmy Digital i Microsoft całe otoczenie systemu ERP. Firma MCI przyjmie na siebie rolę integratora docelowego, kompleksowego rozwiązania informatycznego, zapewniając dodatkowo zarządzanie projektami i działającymi systemami oraz wdrażanie mechanizmów komunikacyjnych. (js)

Serwery HP dla przedsiębiorstw

14 listopada Hewlett-Packard powiadomił o wprowadzeniu do sprzedaży pierwszego serwera z systemem UNIX dla obsługi przedsiębiorstw działających w sieci Internet i sieciach intranetowych. Serwerom towarzyszy oprogramowanie, realizujące funkcje „szersze niż tylko serwera typu Web".

„Nieporównywalna wydajność rodziny serwerów Domain Enterprise Server połączona z różnorodnością oferowanych rozwiązań programowych zapewnia użytkownikom możliwość budowania zaawansowanych systemów informatycznych dla prowadzenia działalności gospodarczej"– twierdzi Ilana Ron, dyrektor HP ds marketingu w zakresie rozwiązań internetowych.

Oprogramowanie instalowane w podstawowych konfiguracjach składa się z następujących modułów:

- HP Domain Business Suite - Pakiet Biznesowy;
- HP Domain Enterprise Server Management System - system zarządzania serwerem, opierający się na interfejsie graficznym OpenView;
- Netscape SuiteSpot - pakiet zapewniający pełne usługi intranetowe włączając w to pocztę HP OpenMail;

Uwaga: klienci mogą zamawiać wymienione oprogramowanie oddzielnie lub zainstalowane w konfiguracji podstawowej. (js)

Oprogramowanie do optymalizacji wykorzystujące Algorytmy Genetyczne

Dariusz Misiurek

Algorytmy Genetyczne (GA) są procedurami poszukiwania, opartymi na przyrodniczych mechanizmach doboru naturalnego oraz dziedziczności. Algorytmy znajdują zastosowanie w wielu dziedzinach nauki, m.in w biologii, informatyce, automatyce, badaniach operacyjnych. Warto jest poznać ich działanie, dlatego w niniejszym opracowaniu zaprezentowano kilka programów, implementujących GA.

Idea algorytmu genetycznego

Idea Algorytmów Genetycznych została zaczerpnięta z przyrody. Naturalną drogą rozwoju żywych organizmów jest ewolucja. Osobniki, które są lepiej przystosowane do życia w pewnym środowisku mają większe szanse na przetrwanie, rozwój, posiadanie potomstwa, natomiast gorsze wyginą. Dobrze poznanym przykładem doboru naturalnego jest proces genetyczny. Chromosom jest zbiorem informacji opisujących osobnika, które w kolejnych pokoleniach przekazywane są potomstwu. Za ewolucyjny rozwój odpowiadają następujące mechanizmy:

- **selekcja** - lepiej przystosowane osobniki mają większe szanse na przeżycie i posiadanie potomstwa;
- **reprodukcja** - tworzenie potomstwa przez pary osobników. Z dwóch chromosomów rodzicielskich wybierane są ciągi genów, które po zespoleniu tworzą nowy chromosom (nowy osobnik będzie miał po części cechy każdego z rodziców). Taka operacja nazywana jest krzyżowaniem.
- **mutacja** - możliwa jest losowa zmiana wartości genów podczas procesu ewolucji;

Przedstawiony mechanizm jest znacznie uproszczony w stosunku do działań Natury, ale jak się okazuje, wystarczający dla różnych zastosowań, jak np: optymalizacja funkcji, co jest tematem tego opracowania. Wyjaśnimy kilka terminów stosowanych w tej dziedzinie. Zakodowane rozwiązanie to chromosom, który składa się z genów. Wartościami genów są allele, a ich pozycja w chromosomie to locus. Wartość optymalizowanej funkcji dla danego chromosomu to przystosowanie.

W zadaniach wykorzystujących GA pierwszym krokiem jest zakodowanie rozwiązania jako chromosomu. Najczęściej używane i najprostsze jest kodowanie binarne, gdzie chromosom traktuje się jako ciąg bitów. Możliwe jest również zakodowanie genu jako liczby całkowitej lub rzeczywistej. Algorytm genetyczny ma wygenerować drogą ewolucji pewnej początkowej populacji rozwiązań (najczęściej losowej) osobnika reprezentującego optymalne rozwiązanie (najlepiej przystosowanego).

Ogólny schemat algorytmu jest następujący:

```
Inicjalizacja populacji osobników (rozwiązań zadania);
Dopóki nie spełniony warunek zatrzymania;
{
        wybór osobników do puli rodzicielskiej na
        podstawie ich przystosowania - selekcja;
        generacja nowego pokolenia - rekombinacja;
        zastąpienie starego pokolenia nowym -
        sukcesja;
}
```

Warunkiem zatrzymania może być: osiągnięcie optymalnego rozwiązania, wygenerowanie określonej ilości pokoleń, osiągnięcie wymaganej zbieżności czy ilości obliczeń funkcji kryterialnej. Wybór osobników do puli rodzicielskiej oraz rekombinacja i sukcesja mogą mieć wiele wariantów, które zostaną omówione w trakcie prezentacji poszczególnych programów. Zainteresowani głębszym poznaniem algorytmów genetycznych mogą skorzystać z [1].

W niniejszym opracowaniu przedstawione zostanie pięć różnych pakietów: od prostych dających niewielkie możliwości (GAC, SGA) do bardziej złożonych, uniwersalnych dla wielu zastosowań (LibGA, Sugal), jak i specjalizowanych (Genocop). Dla wszystkich programów autorzy udostępnili kod źródłowy (Kernighan-Ritchie C lub ANSI C). Dostępna jest też mniej lub bardziej udana dokumentacja do każdego z nich. Na tej podstawie zostały opisane pewne istotne szczegóły implementacyjne każdego z programów. Szczególnie zwrócono uwagę na różne metody inicjalizacji, selekcji, rekombinacji, sukcesji i zatrzymania algorytmu.

Algorym genetyczny w języku C

GAC (Genetic Algorithm in C) jest przykładem bardzo prostego programu implementującego algorytm genetyczny. Autorem GAC jest W.M.Spears z Navy Center for Applied Research in Artificial Intelligence. Program jest napisany w języku C (Kernighan-Ritchie C). W programie chromosom jest zakodowany jako ciąg liczb całkowitych bez znaku, które mogą przyjmować wartości 0 lub 1,

więc można traktować chromosom jako ciąg bitów. Populacja jest tablicą chromosomów. Zarówno maksymalna długość chromosomu, jak i maksymalna wielkość populacji jest stała. Zrezygnowano z dynamicznego przydziału pamięci dla tych struktur w celu uproszczenia oprogramowania. Również wartości prawdopodobieństwa mutacji i krzyżowania są zapisane na stałe w kodzie programu. Aby uruchomić zaimplementowany algorytm dla danego zadania optymalizacyjnego, należy dokonać kilku zmian w kodzie źródłowym. Algorytm genetyczny jest typowy.

Selekcja osobników do populacji reprodukcyjnej odbywa się na podstawie normalizowanej wartości funkcji przystosowania w następujący sposób:

$$f' = \frac{f - f_{avg}}{f_{avg}^{k}} \quad \text{jeśli } f' < 0 \text{ to osadź Equation.2}$$

gdzie:

f - wartość funkcji przystosowania (funkcji kryterialnej), która musi być zawsze dodatnia (musi o to zadbać programista);

f' - znormalizowana wartość funkcji przystosowania;

f_{avg} - średnia wartość przystosowania dla wszystkich generacji populacji (począwszy od populacji początkowej);

f_{avg}^{k} - średnia wartość przystosowania dla bieżącej (k-tej) populacji.

Selekcja odbywa się metodą ruletki, tzn. do puli rodzicielskiej osobniki są losowane z prawdopodobieństwem proporcjonalnym do ich znormalizowanego przystosowania. Widać z tego, że rozwiązania z przystosowaniem mniejszym od średniego nie będą brały udziału w reprodukcji. Dostępne są operatory krzyżowania jednopunktowego, dwupunktowego oraz wielopunktowego i równomiernego. Krzyżowanie jednopunktowe odbywa się w następujący sposób: losowany jest dowolny punkt w chromosomach rodzicielskich, który dzieli każdy z nich na dwie części. Osobniki potomne dziedziczą po jednej części z każdego chromosomu rodzicielskiego. Krzyżowanie wielopunktowe odbywa się analogicznie (wymieniane są odcinki chromosomów wyznaczone przez *n* punktów podziału). Mutacja polega na losowym odwracaniu każdego bitu w każdym chromosomie populacji. Wartości prawdopodobieństwa krzyżowania (0.6) i mutacji (0.001) są zapisane w pliku nagłówkowym *header.h*.

Oprogramowanie zostało stworzone do rozwiązywania problemów maksymalizacyjnych. Aby sprawdzić działanie algorytmu dla własnego zadania, należy stworzyć funkcję obliczającą przystosowanie *(myeval.c)* oraz funkcję sprawdzającą warunek zatrzymania *(term.c)*. Warunkiem zatrzymania może być osiągnięcie zadanego optimum, wygenerowanie zadanej liczby pokoleń, osiągnięcie zadanej zbieżności. Należy pamiętać o tym, że funkcja przystosowania operuje na poszczególnych bitach chromosomów i efektywna implementacja dla nietrywialnego problemu optymalizacyjnego jest złożona. Podczas działania algorytmu obliczane są dwa wskaźniki efektywności:

on-line - średnia wartość funkcji przystosowania ze wszystkich pokoleń dla każdego osobnika;

off-line - średnia wartość funkcji przystosowania najlepszych osobników ze wszystkich pokoleń;

Można wymusić zapis wyników działania algorytmu do plików poprzez odkomentowanie odpowiednich fragmentów kodu *(run.c, geval.c)*. Poważną niedogodnością jest brak dokumentacji do oprogramowania, co w powiązaniu z dosyć ubogimi komentarzami w treści plików źródłowych powoduje znaczne trudności w dostosowaniu oprogramowania do własnych potrzeb (np. modyfikacji operatorów krzyżowania, czy metod selekcji). Wszelkie zmiany wymagają ponownej kompilacji plików źródłowych.

Prosty algorytm genetyczny

SGA (Simple Genetic Algorithm) jest kolejnym przykładem programu implementującego algorytm genetyczny. SGA jest oparty na pascal'owym oprogramowaniu, opisanym w książce D.Goldberga „Genetic algorithms, search, optimization and machine learning". Autorami SGA są R.E.Smith, J.A.Earickson (University of Alabama) oraz D.E.Goldberg (University of Ilinois). Oprogramowanie jest napisane w języku C (Kernighan-Ritchie C) i stanowi dobre uzupełnienie do wydanej niedawno przez WNT książki D.E.Goldberga „Algorytmy genetyczne i ich zastosowania". Razem z oprogramowaniem dostępna jest skromna dokumentacja. Ponadto w plikach źródłowych znajdują się obfite komentarze, które ułatwiają poznanie działania programu.

W SGA każdy osobnik populacji jest pamiętany jako struktura zawierająca: chromosom zaimplementowany jako ciąg bitów, wartość funkcji przystosowania, położenie punktu krzyżowania, informacja o osobnikach rodzicielskich. Krzyżowanie jest jednopunktowe. Mutacja polega na losowym odwróceniu każdego bitu w chromosomie z pewnym prawdopodobieństwem. Selekcja może odbywać się metodą koła ruletki *(plik: rselect.c)*, koła ruletki z wartością oczekiwaną *(plik: srselect.c)*, turniejową *(tselect.c)*. Poniżej opisano te metody.

Metoda koła ruletki

Osobnik jest wybierany do populacji rodziców z prawdopodobieństwem równym ilorazowi wartości jego funkcji przystosowania do sumy wartości funkcji przystosowania wszystkich osobników populacji. Im większa jest wartość jego funkcji przystosowania względem przystosowania innych osobników, tym większa szansa na wylosowanie do populacji reprodukcyjnej. Wszystkie wartości funkcji przystosowania powinny być dodatnie.

Metoda koła ruletki z wartością oczekiwaną

Prawdopodobieństwo wylosowania do populacji reprodukcyjnej osobnika jest liczone w taki sam sposób, jak w me-

Spis treści rocznika INFORMATYKA 1996

ARTYKUŁY PROBLEMOWE

	nr	s.
ABRAMOWICZ WITOLD: Retoryka hipertekstu	3	4
ABRAMOWICZ WITOLD, BARTOSZEWICZ GRZEGORZ, MACIASZEK LESZEK A.: Relacyjne a obiektowe bazy danych - rywalizacja czy koegzystencja?	9	17
ADAMCZEWSKI PIOTR: Wybór systemu informatycznego zarządzania	6	17
BARTOSZEWICZ GRZEGORZ, ABRAMOWICZ WITOLD, MACIASZEK LESZEK A.: Relacyjne a obiektowe bazy danych - rywalizacja czy koegzystencja?	9	17
BASZTURA CZESŁAW: Głosowa komunikacja człowiek-komputer w języku naturalnym	3	9
BOJANOWICZ DARIUSZ: Zintegrowany pakiet CAD do symulacji uszkodzeń w samo- i łatwotestowalnych układach cyfrowych	6	10
BOLC LEONARD, MYKOWIECKA AGNIESZKA, MARCINIAK MAŁGORZATA, KUPŚĆ ANNA: Opis języka naturalnego za pomocą formalizmów unifikacyjnych	3	18
BONI PIOTR: Rozproszone środowisko - narodziny, wiek niemowlęcy	6	19
BORCHERDS PETER H.: Sztuka fizyki obliczeniowej	11	18
BRZEZIŃSKI KRZYSZTOF, ŚREDNIAWA MAREK: Język programowania CHILL	12	23
CICHOSZ PAWEŁ: Wprowadzenie do systemów uczących się ze wzmocnieniem	4	4
COMPLAK WOJCIECH, NAWROCKI JERZY R.: Real-Time Euclid: Język programowania systemów silnie uwarunkowanych czasowo	10	29
CYPRYJAŃSKI JACEK, DUXBURY LINDA: Akademicka informatyka ekonomiczna w Kanadzie - dydaktyka i badania naukowe	6	24
DOLIŃSKA MAŁGORZATA: CIM - kierunek rozwoju przedsiębiorstwa przyszłości	1	18
DUXBURY LINDA, CYPRYJAŃSKI JACEK: Akademicka informatyka ekonomiczna w Kanadzie - dydaktyka i badania naukowe	6	24
FRIH NACEUR: Metody szyfrowania informacji	8	8
FRIH NACEUR: Warunki bezpieczeństwa danych w sieci	11	24
FUGLEWICZ PIOTR, MIŁOSZ MAREK: Informatyczne stopnie zawodowe i komputerowe prawo jazdy użytkownika	5	36
GOŁĘBIEWSKI MICHAŁ: Współpraca pomiędzy aplikacjami w Microsoft Office Professional dla Windows 95	5	32
GOŁUCHOWSKI JERZY, KANIA KRZYSZTOF: Zagadnienia reprezentacji czasu w klasycznych systemach baz danych	7	20
GOŁUCHOWSKI JERZY, KANIA KRZYSZTOF: Reprezentacja czasu w modelu konceptualnym SI	8	5
GOŁUCHOWSKI JERZY, KANIA KRZYSZTOF: Systemy temporalnych baz danych	9	27
GIBBON DAFYDD: Technologia mowy: infrastruktura i zastosowania	3	22
GORAWSKI MARCIN, KONOPACKI ANDRZEJ: MAGIC - kompendium wiedzy	5	18
GÓRSKI JANUSZ, GURBIEL EWA, JURKOWLANIEC MAREK, ZALEWSKI JANUSZ: Nauczanie inżynierii oprogramowania	11	21
GURBIEL EWA, GÓRSKI JANUSZ, JURKOWLANIEC MAREK, ZALEWSKI JANUSZ: Nauczanie inżynierii oprogramowania	11	21
HAMUDA GRZEGORZ: Metody analizy i specyfikacji wymagań. Analiza obiektowa w specyfikacji wymagań systemów czasu rzeczywistego	1	25
HNATKOWSKA BOGUMIŁA, HUZAR ZBIGNIEW: Metoda projektowania HOOD	7	27
HOŁODNIK-JANCZURA GRAŻYNA: Jak płacić programistom?	9	22
HUZAR ZBIGNIEW, HNATKOWSKA BOGUMIŁA: Metoda projektowania HOOD	7	27
HUZAR ZBIGNIEW: Języki programowania czasu rzeczywistego	10	28
JENDRZEJCZYK BEATA, SKOMOROWSKI MAREK: Zastosowanie komputera do automatycznego układania rozkładu zajęć dla szkoły wyższej	4	2
JURKOWLANIEC MAREK, NOWICKI BARTOSZ: Metoda OMT w projektowaniu systemów czasu rzeczywistego	6	34

	nr	s.
JURKOWLANIEC MAREK, GÓRSKI JANUSZ, GURBIEL EWA, ZALEWSKI JANUSZ: Nauczanie inżynierii oprogramowania	11	21
KANIA KRZYSZTOF, GOŁUCHOWSKI JERZY: Zagadnienia reprezentacji czasu w klasycznych systemach baz danych	7	20
KANIA KRZYSZTOF, GOŁUCHOWSKI JERZY: Reprezentacja czasu w modelu konceptualnym SI	8	5
KANIA KRZYSZTOF, GOŁUCHOWSKI JERZY: Systemy temporalnych baz danych	9	27
KOLANO MICHAŁ, SZEDEL JACEK: Analiza specyfiki programowania obiektowego w środowisku implementacyjnym SQL Windows	7	6
KONOPACKI ANDRZEJ, GORAWSKI MARCIN: MAGIC - kompendium wiedzy	5	18
KRISCHER JOSCH: Ewolucja metod zarządzania pamięcią w dużych systemach komputerowych	4	15
KUBOSZEK PIOTR, WIECHECKI PAWEŁ: Wykorzystanie strukturalnych technik CASE w analizie systemów informacyjnych	9	15
KUPŚĆ ANNA, BOLC LEONARD, MYKOWIECKA AGNIESZKA, MARCINIAK MAŁGORZATA: Opis języka naturalnego za pomocą formalizmów unifikacyjnych	3	18
KUPŚĆ ANNA, MARCINIAK MAŁGORZATA, MYKOWIECKA AGNIESZKA: Komputerowe przetwarzanie języka naturalnego	7	1
KWIATKOWSKI JAN, SZEJKO STANISŁAW: Metody projektowania oprogramowania. Programowanie strukturalne systemów czasu rzeczywistego (1)	4	24
KWIATKOWSKI JAN, SZEJKO STANISŁAW: Metody projektowania oprogramowania. Projektowanie strukturalne systemów czasu rzeczywistego (2)	5	40
LASEK MIROSŁAWA: ARIS-Toolset - narzędzia dla Businees Process Reengineering	2	9
LEWOC JÓZEF B., SYDOR ANTONI: Jak budować system Computer Integrated Manufacturing? Technologia	12	13
LIS TOMASZ, SOBKOWIAK ARTUR, STROIŃSKI MACIEJ: Integracja zarządzania sieciami IPX/SPX i TCP/IP (1)	1	8
LIS TOMASZ, SOBKOWIAK ARTUR, STROIŃSKI MACIEJ: Integracja zarządzania sieciami IPX/SPX i TCP/IP (2)	2	18
ŁAWRYNOWICZ ANNA: Implementacja aplikacji narzędziami CASE firmy Oracle	5	24
MACIASZEK LESZEK A., ABRAMOWICZ WITOLD, BARTOSZEWICZ GRZEGORZ: Relacyjne a obiektowe bazy danych - rywalizacja czy koegzystencja?	9	17
MAEGAARD BENTE, STROERUP KARSTEN: Tłumaczenie maszynowe: tendencje rozwojowe i perspektywy dla użytkowników	3	26
MARCINIAK MAŁGORZATA, BOLC LEONARD, MYKOWIECKA AGNIESZKA, KUPŚĆ ANNA: Opis języka naturalnego za pomocą formalizmów unifikacyjnych	3	18
MARCINIAK MAŁGORZATA, KUPŚĆ ANNA, MYKOWIECKA AGNIESZKA: Komputerowe przetwarzanie języka naturalnego	7	1
MARCINKIEWICZ JERZY: Metody obiektowe w analizie i projektowaniu zastosowań informatycznych	10	21
MICHALAK STANISŁAW: GANDALF stawia na rozwój sieci dostępowych i ISDN	8	17
MICHALAK STANISŁAW: GANDALF umożliwia dostęp do Internetu	10	35
MIŁOSZ MAREK, FUGLEWICZ PIOTR: Informatyczne stopnie zawodowe i komputerowe prawo jazdy użytkownika	5	36
MIŁOSZ MAREK: Model edukacji informatycznej społeczeństwa	7	16
MIŁOSZ MAREK, WIT BOGDAN: Programowanie obiektowe w CA-Clipper	8	12
MIKANIK WOJCIECH: Charakterystyka języka OCCAM	11	29
MISIUREK DARIUSZ: Oprogramowanie do optymalizacji wykorzystujące Algorytmy Genetyczne	12	17
MYKOWIECKA AGNIESZKA, BOLC LEONARD, MARCINIAK MAŁGORZATA, KUPŚĆ ANNA: Opis języka naturalnego za pomocą formalizmów unifikacyjnych	3	18
MYKOWIECKA AGNIESZKA, MARCINIAK MAŁGORZATA, KUPŚĆ ANNA: Komputerowe przetwarzanie języka naturalnego	7	1
NAWROCKI JERZY R., COMPLAK WOJCIECH: Real-Time Euclid: Język programowania systemów silnie uwarunkowanych czasowo	10	29
NOWICKI BARTOSZ, JURKOWLANIEC MAREK: Metoda OMT w projektowaniu systemów czasu rzeczywistego	6	34
PAŃKOWSKA MAŁGORZATA: OUTSOURCING - alternatywa zarządzania systemem informatycznym przedsiębiorstwa	6	7
PAŃKOWSKA MAŁGORZATA: Strategia informatyzacji firmy	11	10
PIASECKI MACIEJ: Koncepcja modelu dla obiektowej bazy danych	7	12
PIASECKI MACIEJ: OB! OK! - obiektowy system zarządzania bazą danych	8	1
POLAKOWSKI MACIEJ: Generator aplikacji VisualGen	1	12
POPOŃCZYK ALEKSANDER: Dwa w jednym, czyli system informatyczny i system MRPII w przedsiębiorstwie	10	11
POPOŃCZYK ALEKSANDER: Elementy składowe systemu MRPII	11	6

	nr	s.
POPOŃCZYK ALEKSANDER: Rozważnie, ale i szybko - czyli rzecz o tym, jak wybierać system informatyczny wspomagający MRPII	12	9
PROKOP MAREK: Przegląd cech obiektowego języka PROGRESS 4GL w oprogramowaniu relacyjnych baz danych	5	12
PYTLAK TOMASZ: Programowanie w języku systemów eksperckich CLIPS Cz. 3.: Podstawy programowania obiektowego w języku CLIPS	1	22
Cz. 4.: Integracja systemów eksperckich z programami opracowanymi w innym środowisku	2	22
ROUKENS JAN: Łamanie językowych barier. Ku wielojęzycznemu społeczeństwu informacji w Europie	3	1
RYŻKO JAN: Pamięci podręczne komputerów osobistych	6	15
SACHA KRZYSZTOF M.: Metody projektowania oprogramowania. Projektowanie oprogramowania czasu rzeczywistego	3	38
SKOMOROWSKI MAREK, JENDRZEJCZYK BEATA: Zastosowanie komputera do automatycznego układania rozkładu zajęć dla szkoły wyższej	4	2
SOBKOWIAK ARTUR, LIS TOMASZ, STROIŃSKI MACIEJ: Integracja zarządzania sieciami IPX/SPX i TCP/IP (1)	1	8
SOBKOWIAK ARTUR, LIS TOMASZ, STROIŃSKI MACIEJ: Integracja zarządzania sieciami IPX/SPX i TCP/IP (2)	2	18
SOBOLEWSKI PIOTR: System informacyjny jako przedsięwzięcie	9	11
STAŃCZAK JAROSŁAW: Zagadnienia pozyskiwania wiedzy dla sterowników rozmytych (1)	2	14
STAŃCZAK JAROSŁAW: Zagadnienia pozyskiwania wiedzy dla sterowników rozmytych (2)	4	11
STROERUP KARSTEN, MAEGAARD BENTE: Tłumaczenie maszynowe: tendencje rozwojowe i perspektywy dla użytkowników	3	26
STROIŃSKI MACIEJ, LIS TOMASZ, SOBKOWIAK ARTUR: Integracja zarządzania sieciami IPX/SPX i TCP/IP (1)	1	8
STROIŃSKI MACIEJ, LIS TOMASZ, SOBKOWIAK ARTUR: Integracja zarządzania sieciami IPX/SPX i TCP/IP (2)	2	18
STRYKOWSKI SERGIUSZ: Eksploracja danych	10	17
SYDOR ANTONI, LEWOC JÓZEF B.: Jak budować system Computer Integrated Manufacturing? Technologia	12	13
SZEDEL JACEK, KOLANO MICHAŁ: Analiza specyfiki programowania obiektowego w środowisku implementacyjnym SQL Windows	7	6
SZEJKO STANISŁAW, KWIATKOWSKI JAN: Metody projektowania oprogramowania. Programowanie strukturalne systemów czasu rzeczywistego (1)	4	24
SZEJKO STANISŁAW, KWIATKOWSKI JAN: Metody projektowania oprogramowania. Projektowanie strukturalne systemów czasu rzeczywistego (2)	5	40
SZMUC TOMASZ, SZWED PIOTR: Metody analizy i specyfikacji wymagań. Strukturalno-graficzny język opisu i projektowania aplikacji czasu rzeczywistego	2	26
SZMUC TOMASZ, SZWED PIOTR: Hierarchiczne tworzenie aplikacji czasu rzeczywistego	8	22
SZWED PIOTR, SZMUC TOMASZ: Metody analizy i specyfikacji wymagań. Strukturalno-graficzny język opisu i projektowania aplikacji czasu rzeczywistego	2	26
SZWED PIOTR, SZMUC TOMASZ: Hierarchiczne tworzenie aplikacji czasu rzeczywistego	8	22
SZYJEWSKI ZDZISŁAW: Programowanie obiektowe - nowa jakość programowania	5	3
SZYLLER JERZY: INFORMATYKA w dużych projektach informacyjnych	9	10
SZYLLER JERZY: Ilość czy jakość?	10	9
ŚREDNIAWA MAREK, BRZEZIŃSKI KRZYSZTOF: Język programowania CHILL	12	23
WAWRZONEK LESŁAW: Otwarta droga do programowania obiektowego, czyli OpenROAD/Windows4GL 3.0	5	10
WAWRZONEK LESŁAW: ObjectTeam for OMT	5	30
WIECHECKI PAWEŁ, KUBOSZEK PIOTR: Wykorzystanie strukturalnych technik CASE w analizie systemów informacyjnych	9	15
WIT BOGDAN, MIŁOSZ MAREK: Programowanie obiektowe w CA-Clipper	8	12
WOLSKI PIOTR: Java - programowanie w sieci	5	4
VETULANI ZYGMUNT: Rozumienie języka polskiego przez komputer - założenia metodologiczne	3	30
ZALEWSKI JANUSZ, GÓRSKI JANUSZ, GURBIEL EWA, JURKOWLANIEC MAREK: Nauczanie inżynierii oprogramowania	11	21
ZAMPOLLI ZYGMUNT: Współpraca międzynarodowa w dziedzinie LR	3	34

SYSTEMY INFORMACYJNE

SZYLLER JERZY: INFORMATYKA w dużych projektach informacyjnych	9	10
SOBOLEWSKI PIOTR: System informacyjny jako przedsięwzięcie	9	11
POPOŃCZYK ALEKSANDER: Dwa w jednym, czyli system informatyczny i system MRPII w przedsiębiorstwie	10	11
SZYLLER JERZY: Ilość czy jakość?	10	9
POPOŃCZYK ALEKSANDER: Elementy składowe systemu MRPII	11	6

	nr	s.
PAŃKOWSKA MAŁGORZATA: Strategia informatyzacji firmy	11	10
SYDOR ANTONI , LEWOC JÓZEF B.: Jak budować system Computer Integrated Manufacturing? Technologia	12	13
POPOŃCZYK ALEKSANDER: Rozważnie, ale i szybko - czyli rzecz o tym, jak wybierać system informatyczny wspomagający MRPII	12	9

SPRAWY NAUKI

BORCHERDS PETER H.: Sztuka fizyki obliczeniowej	11	18
GÓRSKI JANUSZ, GURBIEL EWA, JURKOWLANIEC MAREK, ZALEWSKI JANUSZ: Nauczanie inżynierii oprogramowania	11	21

SYSTEMY CZASU RZECZYWISTEGO

CYKL: Metody analizy i specyfikacji wymagań.		
GRZEGORZ HAMUDA: Analiza obiektowa w specyfikacji wymagań systemów czasu rzeczywistego	1	25
SZMUC TOMASZ, SZWED PIOTR: Strukturalno-graficzny język opisu i projektowania aplikacji czasu rzeczywistego	2	26
CYKL: Metody projektowania oprogramowania.		
SZEJKO STANISŁAW, JAN KWIATKOWSKI: Programowanie strukturalne systemów czasu rzeczywistego (1)	4	24
KWIATKOWSKI JAN, SZEJKO STANISŁAW: Projektowanie strukturalne systemów czasu rzeczywistego (2)	5	40
JURKOWLANIEC MAREK, BARTOSZ NOWICKI: Metoda OMT w projektowaniu systemów czasu rzeczywistego	6	34
BOGUMIŁA HNATKOWSKA HUZAR ZBIGNIEW: Metoda projektowania HOOD	7	27
TOMASZ SZMUC, PIOTR SZWED: Hierarchiczne tworzenie aplikacji czasu rzeczywistego	8	22
CYKL: Języki programowania czasu rzeczywistego - HUZAR ZBIGNIEW	10	28
COMPLAK WOJCIECH, NAWROCKI JERZY R.: Real-Time Euclid: Język programowania systemów silnie uwarunkowanych czasowo	10	29
MIKANIK WOJCIECH: Charakterystyka języka OCCAM	11	29
ŚREDNIAWA MAREK, BRZEZIŃSKI KRZYSZTOF: Język programowania CHILL	12	23

OPINIE

ŁĄCZKOWSKI RYSZARD: Fetyszyzm i mitologia komputeryzacji	10	40
SZYCH JERZY: 25 lat mikroprocesora czyli po co jemy tę żabę	12	32
SOKOŁOWSKI STEFAN: O mitologii komputeryzacji - moje trzy grosze	12	33

FELIETONY

SZYLLER JERZY: Integracja i partnerstwo	2	I okł.
KARWICKA KRYSTYNA: Kadra naukowa pilnie poszukiwana	4	I okł.
KARWICKA KRYSTYNA: Infosystem'96	5	I okł.
WAWRZONEK LESŁAW: Przed wyborem systemu	6	I okł.
BASZKIEWICZ DANIELA: Codzienny klimat Internetu	6	40
WAWRZONEK LESŁAW: Rada w radę o pracownikach	7	I okł.
BOGUSŁAWSKI LESZEK: Sobota? - odpada	7	40
WAWRZONEK LESŁAW: Dojrzewanie do stopni zawodowych	8	II okł.
PAJĄK ANDRZEJ: Kanon czy rynek? Co ma być wyznacznikiem kształcenia informatyków?	8	31
WAWRZONEK LESŁAW: Najstarsze ale nowoczesne	9	II okł.
WAWRZONEK LESŁAW:Życie sobie, targi sobie	10	II okł.
WAWRZONEK LESŁAW: Tym razem nie o jazzie	10	6
WAWRZONEK LESŁAW: Czy Unia na nas zaczeka?	11	II okł.

WYWIADY

Od teorii do praktyki - z **prof. dr. hab. Janem Madeyem**, prodziekanem Wydziału Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego, rozmawia *Krystyna Karwicka*	4	I okł.
Nowe oblicze HECTORA - z **Włodzimierzem Łukszą**, Prezesem firmy HECTOR, rozmawia *Jerzy Szyller*	6	1
Aplikacja zdrowego rozsądku - z **dr. Markiem Gondzio**, prezesem Stowarzyszenia Polski Rynek Oprogramowania, rozmawiają *Lesław Wawrzonek i Agnieszka Papis*	9	6
Cyklotron na komputer zamienię - z **prof. Markiem Niezgódką**, dyrektorem Interdyscyplinarnego Centrum Modelowania Matematycznego i Komputerowego Uniwersytetu Warszawskiego, rozmawia *Lesław Wawrzonek*	11	16
Izba firm teleinformatycznych - z **dr. Wacławem Iszkowskim**, prezesem Polskiej Izby Informatyki i Telekomunikacji, rozmawia *Lesław Wawrzonek*	12	8

TERMINOLOGIA

***Reengineering* - a po polsku** - *Jerzy Szyller*	9	32
MR P II - kolejny kłopot z nazewnictwem - *Jerzy Szyller*	10	16
Downsizing...upsizing - *Jerzy Szyller*	11	36
Język Java i komputer sieciowy... - *Jerzy Szyller*	12	36

todzie koła ruletki. Dla każdego osobnika, na podstawie tego prawdopodobieństwa, można obliczyć wartość oczekiwaną ilości potomstwa, jakie może wydać, np. 2.6. W tym wypadku w populacji reprodukcyjnej osobnik będzie miał dwóch reprezentantów. Każdy osobnik zostaje skopiowany do populacji reprodukcyjnej tyle razy, ile wynosi część całkowita wartości oczekiwanej ilości jego potomstwa. Części dziesiętne wartości oczekiwanej są wykorzystane w klasycznej metodzie ruletki, aby uzupełnić populację reprodukcyjną do zadanej wielkości.

Metoda turniejowa

Losowana jest z jednakowym prawdopodobieństwem pewna określona ilość osobników z populacji, a następnie wybierany jest najlepszy z nich, który przechodzi do populacji reprodukcyjnej.

Parametry GA takie, jak prawdopodobieństwa krzyżowania i mutacji czy maksymalna ilość generacji, są wczytywane interakcyjnie. W celu zaimplementowania własnego zadania optymalizacyjnego należy przygotować własną funkcję obliczającą przystosowanie - *objfunc()* oraz inne potrzebne funkcje dla tego zadania (przykładowe pliki: *app.c, app1.c, app2.c).* Wyniki działania będą wyświetlane na ekranie w postaci tekstowej. Chromosomy zostały zaimplementowane zdecydowanie odmiennie niż w GAC. Zastosowano dynamiczne tablice typu *unsigned int.* Każdy bit w tej tablicy odwzorowuje bit ciągu kodowego. Taka implementacja pozwala na lepsze wykorzystanie pamięci, ale trochę komplikuje kodowanie funkcji przystosowania. W programie nie uwzględniono żadnej normalizacji funkcji przystosowania; należy też pamiętać o tym, aby jej wartość była dodatnia.

Biblioteka algorytmów genetycznych

LibGA (Library for Genetic Algorithms) jest biblioteką języka C, stworzoną do badań algorytmów genetycznych. Autorem jest A.R.Corcoran (University of Tulsa). Niestety tak, jak w przypadku GAC, nie ma wystarczającej dokumentacji do biblioteki (poza jednowierszowym opisem plików źródłowych), ale kod źródłowy jest przejrzysty i dostatecznie skomentowany. LIbGA jest oprogramowaniem bardziej elastycznym i oferującym większe możliwości od GAC, czy SGA. Uruchomienie algorytmu genetycznego wymaga dostarczenia funkcji przystosowania dla danego problemu optymalizacyjnego - modyfikacji pliku źródłowego *(ga-test.c).* Dokładniejsza znajomość i głębsza ingerencja w kod źródłowy jest zbędna.

LibGA pozwala na testowanie różnych wariantów algorytmów genetycznych dla tego samego zadania optymalizacyjnego poprzez modyfikację zbioru konfiguracyjnego *(przykładowy plik -ga-test.cfg).* Możliwy jest wybór różnych typów danych zapisywanych w chromosomach:

- ciąg bitów;
- liczby całkowite;
- liczby rzeczywiste;
- permutacja liczb całkowitych.

Dla każdego z tych typów zostały opracowane metody inicjalizacji populacji oraz operatory mutacji i krzyżowania, ze szczególnym zwróceniem uwagi na typ permutacyjny.

Inicjalizacja populacji może być losowa lub odczytana z pliku lub z klawiatury. (dany jest przykładowy plik danych dla inicjalizacji - *initpool.dat*). Można ustalić rozmiary chromosomów i wielkość populacji, maksymalną ilości generacji, wartości prawdopodobieństw mutacji i krzyżowania oraz warunek zatrzymania algorytmu. Zakończenie może nastąpić po uzyskaniu pewnej zbieżności (prawie wszystkie osobniki reprezentują to samo rozwiązanie) lub po określonej ilości generacji.

Selekcja może odbywać się metodą koła ruletki, turniejową lub równomierną. Metody koła ruletki i turniejowa działają tak, jak w przypadku SGA. Metoda selekcji równomiernej polega na losowym wyborze osobników z jednakowym prawdopodobieństwem, co sprawia, że przeszukiwanie przestrzeni rozwiązań jest przypadkowe.

Zaimplementowano kilka operatorów krzyżowania:

Dla reprezentacji bitowej chromosomów:

- **proste**: krzyżowanie jednopunktowe (identyczne jak w przypadku SGA i GAC);
- **jednostajne**: krzyżowanie równomierne polegające na tym, że losowo wybrane bity są zamieniane między chromosomami;

Dla permutacji liczb całkowitych jest kilka typów operatorów m.in.: PMX *(partially matched crossover)*, OX *(order crossover)*, CX *(cycle crossover)* - opis działania można znaleźć w [1], str. 184-190.

Mutacja może być wykonana jako losowe odwrócenie bitu (typ bitowy), losowe wygenerowanie wartości genu (typ całkowity, bitowy, rzeczywisty) lub zamiana genów pozycjami w chromosomie (typ permutacyjny).

Możliwe jest zapisanie w zbiorze konfiguracyjnym, czy rozwiązywany jest problem minimalizacji czy maksymalizacji. Można wymusić zachowanie w populacji potomnej najlepszego osobnika z populacji rodzicielskiej - elitaryzm. Wyniki działania programu zapisuje się w pliku albo wyświetla na ekranie w pełnej lub skróconej formie.

Za pomocą LibGA można badać klasyczny algorytm genetyczny lub zmodyfikowany tzw.: z częściową wymianą (steady-state). Modyfikacja polega na zmianie metody selekcji na turniejową oraz warunkowym wstawianiu nowego osobnika do populacji. Nowy osobnik jest wstawiany do populacji, jeśli jest lepszy od najgorszego w populacji. Tylko jeden osobnik jest zastępowany w populacji podczas jednej generacji. W takim algorytmie jest gwarancja, że tylko lepsze rozwiązania będą pojawiały się w kolejnych generacjach. Algorytm ten pozwala na uzyskanie lepszej zbieżności, jest bardziej selektywny i ogranicza obszar poszukiwań.

Program można uruchomić zarówno w systemie UNIX, jak i DOS. W celu ułatwienia implementowania własnych zadań został stworzony wygodny mechanizm rejestracji podprogramu obliczającego przystosowanie. Można też bardzo łatwo stworzyć własne rozszerzenia, np. nowe operatory, metody inicjalizacji, selekcji itd.

Pakiet Sugal

Sugal (SUnderland Genetic ALgorithm) jest dość zaawansowanym pakietem do badań algorytmów genetycznych. Autorem jest Dr A.Hunter (University of Sunderland, England). Razem z programem dostępna jest obfita dokumentacja, w której zawiera się dokładny opis działania zaimplementowanych metod, interfejsu graficznego oraz plików konfiguracyjnych (*Sugal User Manual*), jak i podręcznik programowania, a właściwie opis zaimplementowanych typów danych funkcji języka C i wykorzystania ich dla własnych potrzeb (*Sugal Programming Manual*). Oprogramowanie zostało napisane w ANSI C. Stworzono interfejs graficzny dla środowiska PC Windows oraz X-Windows. Razem z kodem źródłowym znajduje się wiele przykładowych plików konfiguracyjnych oraz plików źródłowych z zaimplementowanymi różnymi problemami optymalizacyjnymi, m.in problemem komiwojażera. W oprogramowaniu zakodowano wiele metod wykorzystywanych w GA. Program jest doskonałym narzędziem edukacyjnym.

Normalizacja funkcji przystosowania w istotny sposób wpływa na szybkość zbieżności algorytmu genetycznego, ponieważ wszystkie metody selekcji (z wyjątkiem selekcji równomiernej) opierają się na jej wartości. W przypadku, gdy w populacji wszystkie osobniki mają prawie taką samą wartość przystosowania, poszukiwanie staje się mało selektywne, bo osobniki mają równe szanse na posiadanie podobnego do siebie potomstwa. W takim wypadku dobre byłoby „rozciągnięcie" wartości funkcji przystosowania. Z drugiej strony może się też zdarzyć, że w populacji pojawią się osobniki o niezwykle ponadprzeciętnym przystosowaniu. Takie osobniki będą mogły mieć dużo więcej potomstwa od pozostałych, i populacja zostanie szybko przez nie zdominowana. Zawęzi się obszar poszukiwań. Aby tego uniknąć, należy zmniejszyć różnice wartości przystosowania w populacji. Sugal daje możliwość korzystania z kilku metod normalizacji funkcji przystosowania:

- **bezpośrednia**: pewna ustalona liczba jest dodawana do wartości przystosowania każdego chromosomu, a następnie otrzymane wartości są skalowane, aby ich suma dla wszystkich osobników populacji była równa 1.0 (dla zadań maksymalizacji);
- **bezpośrednia odwrotna**: wartość funkcji przystosowania dla każdego chromosomu jest odejmowana od pewnej ustalonej wartości, a następnie otrzymane wartości są skalowane, aby ich suma była równa 1.0 (dla zadań minimalizacji);
- **odwrotna**: do wartości funkcji przystosowania dla każdego chromosomu jest dodawana stała wartość, a następnie brana jest odwrotność wyniku;
- **skalująca**: obliczany jest współczynnik

$$b = \frac{najwieksze_przystosowanie}{najmniejsze_przystosowanie}$$

Następnie przystosowanie jest skalowane w zakresie [1/b, 1] w następujący sposób:

$$f(x) = \frac{1}{b} + (x - m) * \frac{b-1}{M-m}$$

przyst_znormalizowane(x)= f(x)/F;

gdzie:

x - osobnik dla którego przystosowanie znormalizowane jest obliczane;

M - przystosowanie najlepszego w populacji;

m - przystosowanie najgorszego w populacji;

F = suma wartości f(x) dla wszystkich chromosomów w populacji;

- odwrotna skalująca zaproponowana dla problemów minimalizacji działa podobnie jak poprzednio ale

$$b = \frac{najmniejsze_przystosowanie}{najwieksze_przystosowanie}$$

- skalująca wg. średniej oraz odwrotna skalująca wg. średniej działają podobnie do skalujących, ale

$$b = \frac{najwieksze_przystosowanie}{srednie_przystosowanie}$$ - dla skalującej,

$$b = \frac{najmniejsze_przystosowanie}{srednie_przystosowanie}$$ - dla odwrotnej skalującej:

- rankingowa liniowa: należy uszeregować przystosowania od najgorszego: miejsce=0 do najlepszego: miejsce=N-1 (N - ilość osobników w populacji), znormalizowane przystosowanie jest liczone rekurencyjnie:

$$f(x_0) = \frac{2}{N}(B+1);$$

$$f(x_k+1) = f(x_k) + \frac{2(B-1)}{N}(N-1)(B+1);$$

gdzie:

f(x_0) - znormalizowane przystosowanie dla najgorszego osobnika;

f(x_k) i f(x_k+1) - znormalizowane przystosowania dla kolejnych osobników;

B - przystosowanie najlepszego osobnika;

- rankingowa liniowa odwrotna jest wykonywana jak poprzednia, ale uszeregowanie przystosowań od największego: miejsce=0 do najmniejszego: miejsce=N-1 w rankingu;
- rankingowa geometryczna oraz rankingowa geometryczna odwrotna: także należy uszeregować przystosowania (malejąco dla prostej, rosnąco dla odwrotnej normalizacji); dla każdego chromosomu obliczyć:

$$g(x) = m^{f(x)}$$

gdzie $0 < m < 1.0$;

f(x) pozycja osobnika x w rankingu;
g(x) wykładnicza funkcja pozycji osobnika w rankingu;

Znormalizowane przystosowanie dla osobnika x jest równe:

$$\frac{g(x)}{\sum_{x=1}^{N} g(x)}$$ gdzie N - wielkość populacji;

Metody selekcji oparte na znormalizowanej funkcji przystosowania proponowane przez Sugal są takie same jak w poprzednio opisanych pakietach:

- koła ruletki;
- koła ruletki z wartością oczekiwana;
- równomierna;
- turniejowa;

Metody zastępowania starej populacji nową (sukcesja)

- **równomierna**: zastępowana jest cala stara populacja;
- **rankingowa**:
  - **warunkowa:** nowy chromosom przechodzi do populacji, jeśli jest lepszy od najgorszego osobnika populacji;
  - **bezwarunkowa:** pewna ilość chromosomów zastępują równą ilość najgorszych chromosomów populacji starej;
- **turniejowa**:losuje się pewną ilość osobników ze starej populacji, a następnie najgorszy z osobników jest porównywany z osobnikiem z populacji potomnej; jeśli potomny jest lepszy to zastępuje starego;
- **najbardziej podobny**: osobnika z populacji potomnej zastępuje najbardziej podobny do niego osobnik z populacji macierzystej;
- **rodzicielska**: osobniki potomne zastępują osobniki rodzicielskie;

Metody krzyżowania również takie jak w opisanych programach:

- równomierne;
- jednopunktowe;
- dwupunktowe;
- wielopunktowe;

Metody mutacji:

- odwracanie bitowe alleli;
- reinicjacja alleli;
- skokowa zmiana wartości alleli;
- losowa ograniczona skokowa zmiana wartości allelu.
- losowa skokowa zmiana wartości allelu z rozkładem normalnym;

Dwie ostatnie metody są używane w przypadku gdy chromosomy reprezentują typ całkowity lub rzeczywisty.

Jak widać, Sugal daje duże możliwości konstruowania algorytmów genetycznych. Kompletny opis znajduje się w dokumentacji. Kod źródłowy jest przejrzyście i obficie skomentowany, co ułatwia lepsze poznanie działania algorytmów. Interfejs graficzny pozwala na oglądanie wyników działania algorytmu. Można wyświetlić średnią, minimalną i maksymalną wartość funkcji przystosowania, odchylenie standardowe, wariancję. Wyniki działania można zapisać do pliku, a wykresy można wyeksportować w formacie GNUPLT'a. Środowisko graficzne ułatwia konfigurowanie parametrów algorytmu. Sugal pracuje również bez środowiska graficznego. W tym przypadku zmiany parametrów dokonuje się w plikach konfiguracyjnych. Pliki te są używane przez wersje graficzne dla ustawienia wartości początkowych. Jeśli pewien parametr nie został zapisany, to przyjumje się jego wartość domyślną.

Program można uruchomić zarówno w systemie X-Windows, jak i MS Windows. Tak jak dla poprzednich pakietów - implementacja własnego zadania wymaga ingerencji w kod źródłowy. Autor postarał się, aby ingerencja była jak najmniejsza i ograniczała się do modyfikacji jednego pliku źródłowego i utworzenia pliku konfiguracyjnego. W tym celu został stworzony wygodny mechanizm rejestracji podprogramu obliczającego przystosowanie i zbioru konfiguracyjnego.

Można również modyfikować działanie algorytmu poprzez dopisanie własnych metod krzyżowania, selekcji, mutacji, generacji kolejnego pokolenia. Jest to nieco bardziej skomplikowane, ale wyczerpujący opis znajduje się w *Sugal Programming Guide.* Sugal jest bardzo dobrym narzędziem dydaktycznym; daje możliwości pokazania, jak zachowują się różne modyfikacje algorytmów genetycznych i jaki jest wpływ parametrów na działanie.

Pakiet Genocop

Genocop autorstwa Zbigniewa Michalewicza (Department of Computer Science, University of North Carolina) jest specjalistycznym programem do optymalizacji funkcji z ograniczeniami liniowymi. Napisany został w języku C (Kernighan-Ritchie C). Wraz z kodem źródłowym dostępna jest instrukcja obsługi. Niestety brakuje dokumentacji i dobrych komentarzy w kodzie źródłowym, co utrudnia poznanie działania oprogramowania. Jak się wydaje, głównym zamierzeniem autora było stworzenie dobrego i prostego w użyciu narzędzia do optymalizacji funkcji wielu zmiennych z ograniczeniami liniowymi, a nie pokazanie działania algorytmu genetycznego.

Zastosowany algorytm genetyczny nieznacznie odbiega od opisanych poprzednio. Kodowanie jest rzeczywisto

liczbowe. Rekombinacja jest bardziej złożona. Autor zastosował 6 operatorów genetycznych zastępujących klasyczne krzyżowanie i mutację. Operatory w różny sposób modyfikują jednego osobnika (odpowiedniki mutacji) lub parę osobników (odpowiedniki krzyżowania).

Zakodowanie dowolnego zadania optymalizacyjnego jest bardzo proste. Należy zmodyfikować plik źródłowy z funkcją celu (*eval.c*) i stworzyć zbiór konfiguracyjny, w którym zostaną zapisane ewentualne ograniczenia liniowe i kostkowe oraz parametry algorytmu. Program można skompilować zarówno pod systemem UNIX, jak i DOS. Wynikami programu umieszczanymi w pliku są chromosomy najlepszych osobników z kolejnych pokoleń.

Zaprezentowane oprogramowanie prezentuje różny stopień złożoności i przydatności dla własnych zastosowań. Jest on godny polecenia dla tych, którzy chcą zapoznać się z algorytmami genetycznymi i wszystkimi aspektami ich działania, jest pakiet Sugal. Algorytmy genetyczne coraz częściej stają się obiektem zainteresowań ludzi zajmujących się wieloma dziedzinami nauki, dlatego też warto poszukać oprogramowania w innych miejscach intersieci. Zainteresowanie tą problematyką w środowiskach akademickich doprowadziło do powstania na Politechnice Warszawskiej Klubu Algorytmów Ewolucyjnych. W czerwcu tego roku odbyła się w Murzasichle I Krajowa Konferencja Algorytmów Ewolucyjnych. Jednym z jej dokonań było uzgodnienie polskiej terminologii używanej w publikacjach o GA, która została w niniejszym opracowaniu zastosowana.

Oprogramowanie, zaprezentowane w niniejszym artykule, może stanowić uzupełnienie do bogatej literatury i ułatwić poznawanie tej dziedziny. Prezentowane programy są dostępne w archiwum Navy Center for Applied Research in Artificial Intelligence. Można je otrzymać poprzez WWW (http://www.aic.nrl.navy.mil).

Literatura:

[1] GOLDBERG D.E.: Algorytmy genetyczne i ich zastosowania, WNT 1995,
[2] Materiały z I Krajowej Konferencji Algorytmów Ewolucyjnych.

Dariusz Misiurek jest pracownikiem
Instytut Automatyki i Informatyki Stosowanej
Politechniki Warszawskiej
e-mail: dmisiure@neutron.elka.pw.edu.pl

Warunki prenumeraty na 1997 r.

1. Zamówienia na prenumeratę czasopism wydawanych przez Wydawnictwo SIGMA-NOT można składać w dowolnym terminie. Mogą one obejmować dowolny okres czasu, tzn. dotyczyć dowolnej liczby kolejnych zeszytów każdego czasopisma.

2. Wpłat na prenumeratę można dokonywać na ogólnie dostępnych blankietach w Urzędach Pocztowych (przekazy pieniężne) lub Bankach (polecenie przelewu), przekazując opłaty na adres:

Wydawnictwo SIGMA-NOT Spółka z o.o.
Zakład Kolportażu
00-716 Warszawa, skr. pocztowa 1004

konto:
PBK S.A. III O/Warszawa
nr 370015-1573-2720-3-67

Wpłaty na prenumeratę przyjmuja także wszystkie urzędy pocztowe nadawczo-odbiorcze oraz doręczyciele na terenie całego kraju.

3. Prenumerata ulgowa dotyczy członków stowarzyszeń naukowo-technicznych zrzeszonych w FSNT, członków PTI oraz uczniów szkół średnich oraz studentów szkół wyższych. Blankiet wpłaty na prenumeratę ulgową musi być opatrzony na wszystkich odcinkach pieczęcią koła SNT, PTI lub szkoły.

4. Cennik prenumeraty: normalna roczna – 66,00 zł, ulgowa – 33,00 zł
Cena 1 egzemplarza Informatyki wynosi 5,50 zł, ulgowa 2,75 zł.

5. Wszystkie pytania i wątpliwości prosimy kierować do Zakładu Kolportażu:
ul. Bartycka 20, paw. B
00-950 Warszawa
tel.: 40-30-86

SYSTEMY CZASU RZECZYWISTEGO

REDAGUJE ZESPÓŁ: dr inż. Zbigniew FRYŹLEWICZ (Politechnika Wrocławska), prof. dr hab. inż. Janusz GÓRSKI (Francusko-Polska Wyższa Nowych Technik Informatyczno-Komunikacyjnych w Poznaniu), prof. dr hab. inż. Zbigniew HUZAR (Politechnika Wrocławska), prof. dr hab. inż. Tomasz SZMUC (Akademia Górniczo-Hutnicza), dr inż. Kazimierz WORWA (Wojskowa Akademia Techniczna), mgr inż. Zdzisław ŻURAKOWSKI (Instytut Automatyki Systemów Energetycznych we Wrocławiu) - przewodniczący zespołu.

ROK III • GRUDZIEŃ • 1996 • 24

Języki programowania

Język programowania CHILL

Marek Średniawa, Krzysztof Brzeziński

Obserwacja drogi ewolucji systemów telekomunikacyjnych w ciągu ostatnich 30 lat pozwala zauważyć wzrost roli oprogramowania, jako jednego z głównych czynników rozwoju i wzbogacania usług w sieciach publicznych i prywatnych. Zastępowanie urządzeń i systemów o sztywnym sterowaniu (elektromechaniczne centrale telefoniczne, krotnice, systemy transmisyjne) przez nowe, cyfrowe systemy o sterowaniu programowym - SPC (Stored Programme Control) umożliwiają przejście od tradycyjnych, podstawowych usług telefonicznych - POTS (Plain Old Telephone Services), do usług uniwersalnej komunikacji osobistej - PCS (Personal Communication Services).

CHILL spełnia podstawowe wymagania stawiane językom programowania systemów czasu rzeczywistego.

Stosowanie techniki cyfrowej i sterowania programowanego pozwala wbudować „inteligencję" w elementy sieci i urządzenia końcowe. Dzięki temu możliwe jest oferowanie usług o profilach dostosowanych do indywidualnych wymagań użytkowników, zarówno dla abonentów sieci stacjonarnej, jak i dla abonentów ruchowych.

Do usług, których realizacja jest możliwa dzięki oprogramowaniu, należą m.in. klasyczne usługi sieci inteligentnej: połączenie bezpłatne, wirtualna sieć wydzielona, teleglosowanie, numer uniwersalny, poczta wiadomości fonicznych, połączenie kredytowane oraz łączność bezprzewodowa dla abonentów ruchomych (system GSM i inne systemy komórkowe).

Realizacja usług jest tylko jednym z aspektów stosowania oprogramowania w telekomunikacji. Oprogramowanie jest także wykorzystywane do wspomagania wykonywania funkcji zarządzania, eksploatacji i utrzymania, kluczowych dla działania węzłów komutacyjnych i transmisyjnych oraz całej sieci telekomunikacyjnej jako systemu. Należy przy tym podkreślić, że systemy zarządzania całą siecią telekomunikacyjną i jej zasobami są realizowane przez bardzo złożone specjalizowane oprogramowanie.

Oprogramowanie typowej centrali dla sieci publicznej liczy zwykle kilka milionów wierszy kodu źródłowego w języku wysokiego poziomu, np. w CHILL'u. Realizuje ono, oprócz funkcji komutacyjnych, również funkcje zarządzania, eksploatacji i utrzymania (zapewnienie odpowiedniego poziomu jakości usług pomimo występowania uszkodzeń sprzętu i błędów oprogramowania). Zróżnicowany charakter funkcji prowadzi do wzajemnie sprzecznych wymagań. Jest to jedna z głównych przyczyn złożoności oprogramowania, tak w sensie komplikacji, jak i jego rozmiarów. Należy podkreślić, że oprogramowanie realizujące obsługę zgłoszeń - tzw. oprogramowanie komutacyjne - stanowi zwykle tylko niewielką część całości systemu - rzędu kilkunastu procent. Dodatkowym czynnikiem wpływającym na złożoność oprogramowania jest konieczność generowania, eksploatacji i utrzymania wielu wersji i wariantów systemów. Wynika to z konieczności utrzymywania w ruchu i współpracy central różnych generacji technologicznych, pochodzących od wielu producentów i działających w różnym środowisku sieciowym.

Zastosowanie języków wysokiego poziomu do programowania węzłów komutacyjnych, traktowanych jako systemy czasu rzeczywistego, ma zasadnicze znaczenie dla jakości i niezawodności wynikowego oprogramowania. Dzieje się tak przede wszystkim dlatego, że struktury i konstrukcje specjalizowanego języka wysokiego poziomu umożliwiają rozpatrywanie implementowanego systemu komutacyjnego w kategoriach pojęciowych właściwych dla jego specyfiki (mechanizmy komunikacji współbieżnych procesów sekwencyjnych, klasyczne metody dekompozycji strukturalnej i funkcjonalnej, jawne wyrażanie uwarunkowań czasowych i sytuacji wyjątkowych wraz ze specyfikacją sposobu odtwarzania po błędzie).

Ważną ideą jest przyjęcie założenia o sekwencyjności procesów i ich spójności (zachowanie systemu takie, jak-

by każdy proces był wykonywany na przydzielonym mu oddzielnym fizycznym procesorze). Wymaga się także ponadto by działanie oprogramowania było niezależne od względnej szybkości wykorzystywanych procesorów.

CHILL spełnia podstawowe wymagania, sformułowane w pracy [11], stawiane językom programowania systemów czasu rzeczywistego.

Specyfika oprogramowania telekomunikacyjnego

Od systemów telekomunikacyjnych oczekuje się spełnienia typowych wymagań stawianych systemom czasu rzeczywistego:

- współbieżność na dużą skalę (np. kilka tysięcy jednocześnie aktywnych współbieżnych procesów);
- szeroki zakres skali ziarnistości czasu od pojedynczych milisekund do godzin, dni i miesięcy;
- wysoka wydajność systemu warunkująca odpowiedni czas reakcji;
- wysoka niezawodność, stopień gotowości i dostępności systemu, tolerowanie błędów, stopniowa łagodna degradacja jakości usług w przypadku awarii lub błędu.

Zastosowania telekomunikacyjne mają ponadto szereg cech szczególnych odróżniających je od typowych systemów informatycznych. Główne z nich to:

- Złożoność odzwierciedlana przez rozmiar oprogramowania: typowo od 100 000 do 10 000 000 wierszy kodu źródłowego w języku wysokiego poziomu.
- Bardzo długi czas eksploatacji i utrzymania (pielęgnacji) systemu (w porównaniu do typowych aplikacji informatycznych), zwykle od 3 do 20 lat.
- Ciągła ewolucja systemu w trakcie eksploatacji, obejmująca kilka generacji technologii, w sensie zarówno sprzętu, jak i oprogramowania.
- Wymaganie przystosowania do aktualizacji oprogramowania „w ruchu", bez naruszania ciągłości świadczenia usług.
- Rozproszona, sieciowa architektura systemu rozwojowego (goszczącego) i systemu docelowego.
- Liczne zespoły wykonawcze o dużej fluktuacji personelu, z uwagi na rozpiętość czasową procesów implementacji, eksploatacji i pielęgnacji.
- Praca w dużych zespołach projektowych i programistycznych, rozproszonych geograficznie, narzucająca stosowanie metodologii i narzędzi zapewniających integralność i spójność opracowywanego systemu.
- Koszt cyklu życiowego znacznie przekraczający koszty początkowego wdrożenia.
- Konieczność zarządzania wieloma wersjami, wariantami i konfiguracjami oprogramowania podczas generacji wersji początkowej i podczas aktualizacji.
- Konieczność zachowania zgodności z normami międzynarodowymi (ITU-T, ETSI i ISO/IEC) i krajowymi.

Rys historyczny

Inicjatywa opracowania znormalizowanego języka programowania dla zastosowań telekomunikacyjnych została podjęta przez CCITT w roku 1973. Pierwszym krokiem była analiza i ocena stosowanych w praktyce 27 języków programowania, z których wyselekcjonowano grupę 6 języków stanowiących podstawę do dalszych prac. Do wspomnianej grupy wybrano następujące języki:

- DPL: NTT - Japonia;
- ESPL/I: ITT (obecnie Alcatel) - Stany Zjednoczone, Belgia;
- Mary: SINTEF/RUNIT - Norwegia;
- PAPE: France Telecom/CNET - Francja;
- PLEX: Ericsson - Szwecja;
- RTL2: University of Essex - Wielka Brytania;

Analiza własności wymienionych wcześniej języków doprowadziła do wniosku, że żaden z nich nie spełnia postawionych wymagań, i w związku z tym powołano „Zespół specjalistów", któremu powierzono zadanie opracowania nowego języka. W zespole tym uczestniczyli przedstawiciele dużych firm telekomunikacyjnych i operatorów sieci. Wstępna propozycja definicji języka została przedstawiona na forum CCITT w 1976 roku, a język nazwano CHILL (CCITT HIgh Level Language). W następnym okresie studiów, w latach 1977-80, podjęto weryfikację praktyczną

Kamienie milowe rozwoju inżynierii oprogramowania w zastosowaniach telekomunikacyjnych:

1965	Instalacja w sieci publicznej Stanów Zjednoczonych pierwszej centrali telefonicznej o sterowaniu programowym: system ESS No 1 firmy Bell.
1968	Rozpoczęcie prac normalizacyjnych przez CCITT.
1973	Rozpoczęcie prac nad definicją rodziny języków obejmujących pełny cykl życia oprogramowania dla systemów komutacyjnych: języka specyfikacji - SDL, języka programowania - CHILL i języka interakcji operatora z systemem - MML.
1976	Przyjęcie wstępnej propozycji definicji języka programowania CHILL.
1980	Zatwierdzenie przez CCITT definicji języka CHILL jako zalecenia Z.200.
1981-84	Wprowadzenie do języka CHILL mechanizmów kompilacji cząstkowej, rozszerzenie zasad widoczności nazw, definicja modelu i procedur wejścia/wyjścia.
1985-88	Wprowadzenie reprezentacji wartości czasowych i operacji odmierzania czasu. Ustabilizowanie zaleceń serii Z.100 definiujących język specyfikacji SDL.
1989	Zatwierdzenie definicji języka CHILL jako normy ISO/IEC (ISO/IEC Standard 9496).
1989-92	Uzupełnienie zestawu typów predefiniowanych o typ rzeczywisty. Zatwierdzenie definicji języka specyfikacji SDL uwzględniającej mechanizmy podejścia obiektowego.
1993-96	Prace ITU-T (do 1992 CCITT) nad rozszerzeniami obiektowymi języka CHILL.

przedstawionej propozycji. „Zespół specjalistów" przekształcono w „Forum Implementatorów", którego zadaniem było opracowanie kompilatorów języka i zebranie doświadczeń ze stosowania języka. Wynikiem prac była definicja języka CHILL, zatwierdzona przez CCITT w 1980 roku, jako zalecenie Z.200.

Język CHILL odziedziczył wiele cech popularnych języków programowania wysokiego poziomu, przede wszystkim języków PL/I, Pascal i Modula. Na ostateczny kształt języka miały też wpływ doświadczenia z eksploatacji, wymienionych wcześniej, specjalizowanych języków programowania stosowanych przez producentów central. Po 1987 roku opracowano na świecie ponad 30 kompilatorów i środowisk wspomagających programowanie w języku CHILL, w tym dwa w Instytucie Telekomunikacji Politechniki Warszawskiej (IT-CHILL i CHILLIT).

Status języka CHILL na świecie

Użytkownikami języka CHILL są głównie duże firmy telekomunikacyjne produkujące centrale telefoniczne, które były zaangażowane w prace normalizacyjne. Należą do nich między innymi firmy: Alcatel, Daewoo Telecom, Fujitsu, Hitachi, GPT, Italtel, NEC, NTT, Oki, Philips, Siemens, Telebras, Telettra. Z wykorzystaniem języka CHILL zostało zaimplementowane oprogramowanie dla ponad 20 systemów central telefonicznych przeznaczonych dla sieci publicznych, central aboneckich i innych systemów. Wśród nich są systemy znane i eksploatowane na rynku polskim: rodziny central Alcatel S12 i E10, Siemens EWSD, centrala abonencka Siemens Hicom. Według oszacowań [9], około 45% cyfrowych łączy abonenckich zainstalowanych w 1990 roku było obsługiwanych przez oprogramowanie napisane w CHILL'u.

W Instytucie Telekomunikacji język CHILL został wykorzystany do implementacji oprogramowania dla prototypu centrali telefonicznej U10. Ponadto język CHILL jest wykorzystywany w procesie dydaktycznym na Wydziale Elektroniki i Technik Informacyjnych, w zajęciach poświęconych inżynierii oprogramowania (m.in. do kursu „Specyfikacja i oprogramowanie systemów komutacyjnych").

Według szacunków [9], język CHILL jest używany na świecie przez około 15 tysięcy programistów. Dostępność kompilatorów i uniwersalnych środowisk wspomagających dla języka CHILL, na zasadach komercyjnych, ogranicza się do oferty systemu CHIPSY norweskiej firmy Kvatro). Sytuacja ta wynika przede wszystkim ze specyfiki zastosowań, polegającej na ukierunkowaniu środowisk na własne, firmowe architektury systemów docelowych. Dodatkowym czynnikiem jest także ochrona własnej technologii oprogramowania i narzędzi przed konkurencją.

CHILL - prezentacja języka

CHILL jest proceduralnym językiem programowania współbieżnego o strukturze blokowej, z rozbudowanymi mechanizmami kontroli zgodności typów i widoczności nazw. Swoje główne cechy, w części sekwencyjnej, odziedziczył z języków takich, jak Pascal i Modula. Na ostateczny kształt języka miały też wpływ doświadczenia ze stosowania specjalizowanych języków programowania systemów komutacyjnych, takich jak np. PAPE, PLEX i ESPL/I, oraz ówczesny stan wiedzy w dziedzinie inżynierii oprogramowania. W sensie „siły ekspresji", podstawowych pojęć, ogólnych zasad składniowych, pragmatyki stosowania i kategorii systemów, dla których jest przeznaczony, CHILL jest porównywalny z językiem Ada, zaprojektowanym w latach osiemdziesiątych na zlecenie Departamentu Obrony Stanów Zjednoczonych. Główne cechy CHILL'a to:

- Ukierunkowanie na implementację dużych systemów i pracę zespołową. Temu celowi służą mechanizmy dekompozycji systemu na moduły, z separacją części specyfikacyjnej, implementacyjnej oraz kontekstu, umożliwiające kompilację cząstkową programów.
- **Przystosowanie do implementacji rozproszonych systemów czasu rzeczywistego ze współbieżnością.** Służą temu przyjęte mechanizmy asynchronicznej interakcji procesów za pomocą sygnałów, zdarzeń i buforów oraz rozbudowane środki nadzoru czasowego (np. możliwość wyrażania limitów czasu oczekiwania).
- **Położenie nacisku na bierną i aktywną niezawodność programów.** Służą temu celowi zróżnicowane i bogate środki: rozbudowane mechanizmy definiowania i kontroli zgodności typów danych, zasady używania zmiennych roboczych w instrukcjach złożonych, precyzyjna kontrola widoczności nazw, mechanizm obsługi wyjątków umożliwiający programową reakcję na błędy wykonania.
- **Uwzględnienie wymagań dotyczących efektywności kodu wynikowego.** Służą temu celowi mechanizmy jawnej konwersji typów oraz definicji wewnętrznego upakowania (reprezentacji) struktur danych.

W zaleceniu Z.200 stosowana jest terminologia nieco odbiegająca od powszechnie używanego słownika pojęć dla najpopularniejszych języków programowania. W artykule starano się zachować terminologię typową, czyniąc jedynie wyjątek dla pojęć CHILL'a, dla których brakowało odpowiednika.

Struktura programu

W sensie statycznym, *program* w CHILL'u ma hierarchiczną strukturę blokową. Jednostkami najwyższego poziomu są *moduły* (MODULE) i *regiony* (REGION). Oznacza to, że program jest zdefiniowany jako pewien zbiór modułów i regionów (użyto słowa „zbiór" w celu podkreślenia faktu, że trudno mówić o tekstowej kolejności modułów i regionów, charakterystycznej dla programów w takich językach, jak Pascal i C). Moduł stanowi środek dekompozycji funkcjonalnej programu i kontroli widoczności nazw, pozwalający grupować definicje stałych i typów danych, deklaracje zmiennych, definicje procesów i procedur oraz związane z nimi akcje. Specjalna rola modułu polega na definiowaniu interfejsu zewnętrznego, określającego listę nazw obiektów (typów danych, procesów, sygnałów, procedur, zmiennych) udostępnianych przez dany moduł innym modułom. Separacja interfejsu zewnętrznego modułu od jego implementacji jest wyrażana przez syntaktyczne wyróżnienie *części specyfikacyjnej* (MODULE SPEC) i *części implementacyjnej modułu* (MODULE BODY). W części specyfikacyjnej umieszcza się definicje stałych i typów da-

nych, definicje szkieletów procesów i procedur (tylko nagłówki bez instrukcji czynnych), deklaracje sygnałów i zmiennych globalnych, oraz *instrukcje udostępnienia* (eksportu) (GRANT) i *przechwycenia* (importu) (SEIZE) nazw. W myśl przyjętych reguł widoczności, udostępnienie nazwy wymaga jej „wyeksportowania" w części specyfikacyjnej modułu macierzystego oraz jej „zaimportowania" przez części specyfikacyjne modułów korzystających. W celu uporządkowania związków pomiędzy modułami wprowadzono możliwość definiowania tzw. *okien widoczności*, służących do grupowania nazw podlegających wspólnemu eksportowi lub importowi. Wspomniany wcześniej podział umożliwia realizację rozłącznej kompilacji modułów, będącej podstawą pracy zespołowej przy implementacji złożonych i dużych systemów oprogramowania.

Przyjęte zasady dekompozycji fukcjonalnej sprawiają, że działanie pewnych modułów może być uwarunkowane dostępnością operacji i obiektów udostępnianych przez inne moduły. Innymi słowy moduł jest „wykonywalny" w kontekście innych modułów. Do wyrażania wspomnianego uwarunkowania służy w CHILL'u konstrukcja *specyfikacji kontekstu* (CONTEXT).

Region jest szczególną odmianą modułu, służącą do implementacji mechanizmu wzajemnego wykluczania dla procesów korzystających z operacji na zmiennych w nim zadeklarowanych.

Pozostałymi środkami dekompozycji strukturalnej programu są *procesy* (PROCESS), *procedury* (PROC) i *bloki podstawowe*. Proces stanowi podstawową jednostkę wykonania współbieżnego. Procesy mogą być dynamicznie kreowane i unicestwiane.

Wykonujący się program stanowi w sensie dynamicznym zbiór współdziałających ze sobą procesów. Definicja języka przewiduje wiele sposobów interakcji: od jawnej komunikacji asynchronicznej za pomocą sygnałów po współpracę przez zmienne dzielone.

Dane

Danymi w CHILL'u są wartości i zmienne. Wartości są reprezentowane przez stałe (SYNONYM) i literały. Dla pewnych standardowych typów danych zostały wprowadzone predefiniowane literały oznaczające wartości danego typu (np. TRUE i FALSE dla typu logicznego BOOL). Z każdą zmienną związany jest jej typ, określający zbiór dopuszczalnych wartości, zestaw operacji oraz inne własności, takie jak: rozmiar, sposób reprezentacji w pamięci, dostępność dla zapisu, możliwość odwoływania się do zmiennej za pomocą wskazania.

W języku wprowadzono pewien zestaw predefiniowanych typów danych oraz określono zasady konstruowania przez programistę własnych typów danych. W języku wyróżniono następujące kategorie typów:

- Typy dyskretne:
  - całkowity (INT);
  - znakowy (CHAR);
  - logiczny (BOOL);
  - typy wyliczeniowe (SET);
  - typy okrojone (zdefiniowane przez ograniczenie wartości bazowego typu dyskretnego).
- Typy proste:
  - typ rzeczywisty (FLOAT);
  - typ zbiorowy (POWERSET);
  - typ identyfikujący procesy (INSTANCE);
  - typy synchronizacyjne, służące do definiowania buforów (BUFFER) i zdarzeń (EVENT);
  - typy związane z odmierzaniem upływu czasu:
    - typ reprezentujący czas bezwzględny (TIME);
    - typ reprezentujący odcinki czasu (DURATION);
  - typy proceduralne, służące do dynamicznej manipulacji procedurami;
  - typy transmisyjne, służące do obsługi wejścia/wyjścia programu.
  - typy wskazujące:
    - związane (REF) (definiuje wskazania na zmienne określonego typu statycznego, powoływanego w opisie tego typu);
    - swobodne (PTR) (definiuje wskazania na zmienne różnych typów; umożliwia budowanie list złożonych z elementów różnych typów);
    - szeregowe (ROW) (definiuje wskazania na zmienne typów dynamicznych);
- Typy złożone definiujące wartości zbudowane z elementów składowych innych typów, do których można niezależnie się odwoływać. W zależności od sposobu agregacji elementów składowych wyróżniono:
  - typy tablicowe (ARRAY);
  - typy rekordowe (STRUCT) (w tym rekordy z wariantami);
  - typy łańcuchowe; (specjalny rodzaj tablic jednowymiarowych o elementach typu znakowego lub bitowego);

Ważną cechą języka, z punktu widzenia niezawodności oprogramowania, jest możliwość wykorzystywania zmiennych niezapisywalnych (opatrzonych atrybutem READ), inicjalizowanych w momencie deklaracji. W ten sposób można zabezpieczyć przed modyfikacją zmienną złożoną jako całość lub poszczególne jej składniki.

Innym mechanizmem niezawodności biernej jest wprowadzenie dwóch rodzajów definicji typów danych: NEWMODE i SYNMODE. Definicja „nowego typu" NEWMODE umożliwia deklarowanie zmiennych, które z definicji, w zamierzeniu, mają typ niezgodny z innymi zmiennymi o identycznej budowie. Dzięki temu istnieje możliwość zabezpieczenia się przed omyłkowym użyciem identycznie reprezentowanych zmiennych w sposób niezgodny z ich pragmatycznym znaczeniem w programie (np. liczniki typu całkowitego rejestrujące zdarzenia różnych kategorii - alarmy i impulsy zaliczeniowe). Użycie mechanizmu definicji nowego typu sprawia, że opisany rodzaj błędów jest wykrywany w fazie kompilacji programu. Ma to istotny wpływ na koszt poprawienia błędu.

Mechanizm definiowania typów synonimowych SYNMODE reprezentuje klasyczny sposób definiowania typów, znany np. z języka Pascal.

Instrukcje programowania sekwencyjnego

Algorytmiczną część programu tworzą instrukcje czynne, opisujące akcje. Wśród nich można wyróżnić instrukcje programowania sekwencyjnego oraz instrukcje służące wyrażaniu współbieżności i związane z wyjątkami, które omówiono dalej.

Instrukcje programowania sekwencyjnego w CHILL'u są udoskonaloną formą konstrukcji znanych z innych języków programowania strukturalnego (Pascal, Ada, Modula-2).

Instrukcje czynne dzielą się na instrukcje proste i instrukcje złożone. Do instrukcji prostych należą:

- instrukcja przypisania (:=);
- instrukcja wywołania procedury (CALL);
- instrukcja powrotu z procedury (RETURN);
- instrukcja określenia rezultatu procedury funkcyjnej (RESULT);
- instrukcja zaniechania (EXIT);
- instrukcja skoku (GOTO);

Wśród wymienionych instrukcji na uwagę zasługuje instrukcja zaniechania. Służy ona do przerywania, w sposób strukturalny, wykonania zagnieżdżonych instrukcji złożonych (np. zagnieżdżonych pętli).

Instrukcje złożone służą do grupowania i zagnieżdżania akcji oraz definiowania przebiegu sterowania w programie. Dostępne są następujące instrukcje złożone:

- instrukcja warunkowa (IF ... FI);
- instrukcja wyboru (CASE ... ESAC);
- instrukcja pętli (DO ... OD) umożliwiająca opisanie trzech rodzajów iteracji:
  - iteracja sterowana warunkiem logicznym (DO WHILE ... OD);
  - iteracja z licznikiem powtórzeń pętli (DO FOR ...OD);
  - pętla nieskończona (DO FOR EVER ...OD);
- instrukcja wiążąca (DO WITH ...OD) ułatwiająca odwoływanie się do złożonych struktur danych;

Przyjęto zasadę, że nawias składniowy zamykający instrukcję stanowi zapisany wspak nawias otwierający (np. dla instrukcji warunkowej IF ... FI;).

Mając na uwadze wymagania biernej niezawodności oprogramowania w instrukcjach złożonych wprowadzono szereg udoskonaleń.

W instrukcji pętli z licznikiem wprowadzono mechanizm automatycznej deklaracji lokalnej zmiennej licznika powtórzeń pętli. Programista ma możliwość określenia typu licznika, kierunku jego zmiany wartości (inkrementacja lub dekrementacja) i kroku zmiany. Istnieje odmiana instrukcji iteracyjnej, służącej do przeglądania tablic, w której zmienna sterująca reprezentuje kolejne elementy przetwarzanej tablicy. Dopuszcza się łączenie części sterującej dla iteracji z warunkiem logicznym i pętli z licznikiem. Możliwe jest też jednoczesne użycie kilku zmiennych sterujących.

Wyjątki

Jednym z podstawowych wymagań, stawianych systemom nadążnym o działaniu ciągłym (systemom czasu rzeczywistego), jest odporność na błędy oprogramowania i awarie sprzętowe oraz inne niepoprawne lub anormalne stany, powstające w trakcie wykonania. Ich wystąpienie nie może powodować zatrzymania pracy systemu, lecz co najwyżej przejściowe pogorszenie jakości lub dostępności usług i jednoczesne aktywowanie procedur odtwarzania po błędzie, inicjujące powrót do stanu prawidłowego.

W celu realizacji opisanych wymagań wprowadzono w CHILL'u pojęcie *wyjątku* i jego *obsługi*. Rozróżnia się dwa typy wyjątków: standardowe i programowe. Pierwsze są wymuszane niejawnie przez klasyczne błędy wykonania, takie jak np. indeks spoza zakresu (RANGEFAIL), próba wysłania sygnału do nieistniejącego procesu (EXTINCT), wyczerpanie puli pamięci dynamicznej podczas żądania jej przydziału (ALLOCATEFAIL). Nieco inny charakter ma tylko wyjątek standardowy ASSERTFAIL, wymuszany, gdy logiczna funkcja standardowa ASSERT, której parametrem jest pewne wyrażenie logiczne (zwykle wyrażające poprawny stan systemu), zwraca wartość FALSE.

W odróżnieniu od wyjątków standardowych wyjątki programowe są w sposób jawny i zamierzony powodowane przez program, za pomocą intrukcji wymuszenia wyjątku (CAUSE).

Wystąpienie wyjątku w pewnym miejscu programu powoduje przekazanie sterowania do właściwej dla niego sekwencji akcji, stanowiącej jego obsługę. Obsługa wyjątku (ON ... END) może być dołączona do każdej instrukcji języka (prostej i złożonej) oraz do definicji jednostki strukturalnej programu (procedury, procesu, modułu, regionu), w których potencjalnie może zajść wyjątek. Obsługa wyjątku ma budowę analogiczną do instrukcji wyboru, dzięki czemu jest możliwe opisanie w jednym miejscu sposobu reakcji na wszystkie explicite wymienione wyjątki oraz wszystkie inne pozostałe (część ELSE). Po wykonaniu ciągu akcji, stanowiącego obsługę wyjątku, sterowanie jest przekazywane do instrukcji, następującej bezpośrednio po instrukcji, do której była dołączona właściwa obsługa wyjątku. Oznacza to na przykład, że jeśli wyjątek spowodowała pewna instrukcja umieszczona wewnątrz instrukcji złożonej (np. pętli), do której dołączono odpowiednią obsługę, to po obsłużeniu wyjątku wykonanie programu byłoby kontynuowane od pierwszej instrukcji za tą instrukcją złożoną.

Wyszukiwanie właściwej obsługi dla wyjątku odbywa się od miejsca jego wystąpienia (instrukcji), w kierunku na zewnątrz, z uwzględnieniem kolejnych, otaczających instrukcji złożonych, a następnie jednostek strukturalnych hierarchii programu.

Duże znaczenie praktyczne mają wyjątki związane z procedurami, nazywane wyjątkami formalnymi. Lista wyjątków formalnych jest umieszczana w nagłówku procedury (EXCEPTIONS). Wyróżnienie wyjątków formalnych i pozostałych wyjątków pozwala zdecydować, czy wyjątek ma być obsłużony wewnątrz procedury, która spowodowała jego wystąpienie czy też ma być przekazany do punktu wywołania procedury. W tym ostatnim przypadku wybór obsługi zależy nie tylko od warunków statycznych, lecz także od liczby i kolejności wywołania procedur, co pozwala na uwzględnienie kontekstu dynamicznego, w jakim pojawił się wyjątek. Dzięki opisanemu mechanizmowi wymuszenie wyjątku może spowodować zakończenie wykonania

kilku procedur w łańcuchu wywołań oraz procesu, który je wywołał bezpośrednio lub pośrednio.

Opisany sposób wyszukiwania obsługi wyjątków sprawia, że są one bardzo wygodnym środkiem organizacji przebiegu sterowania w programie, pozwalającym z jednej strony na zapewnienie aktywnej niezawodności oprogramowania, a z drugiej na uproszczenie struktury programów.

Warunkiem poprawności semantycznej programu CHILL'owego, w sensie opisanej wcześniej odporności na błędy, jest zapewnienie obsługi dla wszystkich wyjątków, które mogą się pojawić w trakcie jego wykonywania. W celu uniknięcia konieczności jawnego definiowania obsługi dla wszystkich wyjątków mogących wystąpić w programie, w implementacjach (np. w środowisku CHILLIT [7]) przyjęto mechanizm domyślnej obsługi dla wyjątków standardowych. Na przykład, dla procesu przyjęto, że domyślna obsługa polega na zakończeniu procesu i zwróceniu zasobów przez niego wykorzystywanych.

Mechanizmy współbieżności

Podstawową jednostką wykonania współbieżnego jest proces sekwencyjny. Program, w sensie dynamicznym, jest zbiorem procesów, które w różny sposób mogą ze sobą współpracować. W danej chwili wiele procesów może być wykonywanych współbieżnie. W statycznej strukturze programu procesy są wyróżnione przez definicje. Definicja procesu opisuje jego obiekty lokalne (zmienne, procedury, wyjątki) oraz algorytm jego działania. Na podstawie jednej definicji może zostać powołanych do życia wiele wystąpień procesów.

Język CHILL umożliwia dynamiczne kreowanie i unicestwianie procesów. Wystąpienie procesu jest tworzone w wyniku wykonania instrukcji START przez proces kreujący. Zawiera ona nazwę definicji procesu oraz jego parametry. Proces jest unicestwiany w wyniku wykonania instrukcji STOP lub wskutek osiągnięcia przez sterowanie nawiasu syntaktycznego zamykającego proces. Wystąpienia procesów są identyfikowane za pomocą wartości typu INSTANCE. Tożsamość wystąpienia procesu może być przekazana: procesowi kreującemu w instrukcji kreacji (START), procesowi odbierającemu sygnał w instrukcji odbioru (RECEIVE CASE) lub uzyskana w wyniku użycia operatora THIS.

Wymagania związane z implementacją systemów komutacyjnych sprawiły, że w CHILL'u przewidziano kilka podstawowych mechanizmów współdziałania procesów współbieżnych. Mechanizmy te różnią się pomiędzy sobą stopniem jawności i szczegółowości kontroli koordynacji procesów przez programistę:

- **Synchronizacja** polega na wstrzymywaniu wykonania procesu w oczekiwaniu na osiągnięcie pewnego punktu lub stanu przez inny proces albo zajście zdarzenia zewnętrznego. Środkiem synchronizacji procesów są zmienne typu zdarzeniowego (EVENT) i wykorzystujące je instrukcje wstrzymania (DELAY) i wznowienia (CONTINUE). Proces, który wykonał instrukcję DELAY zostaje wstrzymany aż do chwili, kiedy inny proces wykona instrukcję CONTINUE, odnoszącą się do zdarzenia synchronizującego. Instrukcja wznowienia nie powoduje żadnych skutków, jeżeli żaden z procesów nie oczekuje na dane zdarzenie. Proces może oczekiwać na zajście jednego z wielu zdarzeń. Za pomocą instrukcji wstrzymania z wyborem (DELAY CASE) można dla każdego ze zdarzeń określić odmienny sposób kontynuacji programu. Zmienne typu zdarzeniowego reprezentują kolejkę procesów oczekujących na pewne zdarzenie. Długość tej kolejki może zostać określona w deklaracji zmiennej typu zdarzeniowego. Instrukcje wstrzymania i wznowienia użyte w procedurach krytycznych regionu powodują odpowiednio jego zwalnianie i zajmowanie.
- **Komunikacja** polega na jawnej wymianie wiadomości przekazujących dane między procesami. Komunikacja może mieć charakter bezpośredni i jawny (wiadomości kierowane są jawnie od procesu do procesu) lub pośredni, gdy wiadomości są przekazywane przez bufor - „skrzynkę pocztową" między grupą procesów „nadawców" do grupy procesów „odbiorców". Komunikacja pociąga za sobą synchronizację procesów. Środkiem bezpośredniej komunikacji procesów są sygnały i wykorzystujące je instrukcje nadania i odbioru sygnałów. Sygnały są wprowadzane do programu za pomocą deklaracji SIGNAL, określającej nazwę sygnału, typ przenoszonych danych oraz ewentualnie nazwę procesu, którego wystąpienia mogą być odbiorcami sygnału. Do wysyłania sygnałów służy instrukcja nadania (SEND), która oprócz wartości danych może jawnie identyfikować wystąpienie procesu adresata. Instrukcja odbioru (RECEIVE CASE) pozwala oczekiwać na wiele sygnałów i uzależniać kontynuację wykonania programu od odbioru konkretnego sygnału. Komunikacja procesów za pomocą sygnałów ma charakter asynchroniczny. W odróżnieniu od zdarzeń sygnały są trwałe. Środkiem komunikacji pośredniej są zmienne typu buforowego (BUFFER) i związane z nimi instrukcje nadania komunikatu do bufora (SEND) i odbioru komunikatu z bufora (RECEIVE). Procesy wysyłają do siebie komunikaty za pomocą wspólnego bufora, który pełni rolę skrzynki pocztowej. Pojemność bufora może być ograniczona. Jeżeli jest miejsce w buforze, proces nadawca nie jest wstrzymywany, w przeciwnym razie proces jest wstrzymywany do momentu zwolnienia się miejsca w buforze, w wyniku odebrania komunikatu przez jeden z procesów oczekujących.
- **Wzajemne wykluczanie** jest stosowane w celu zapewnienia bezpiecznego - chroniącego przed niepożądaną interferencją - współużytkowania wspólnych zmiennych - zasobów przez wiele procesów. Wzajemne wykluczanie pociąga za sobą synchronizację procesów. Środkiem realizacji wzajemnego wykluczania są regiony, które oprócz definicji typów i deklaracji zmiennych mogą zawierać tylko definicje procedur działających na tych zmiennych. Procedury udostępniane przez region są nazywane *procedurami krytycznymi*. Proces wywołujący procedurę krytyczną jest zawieszany, jeżeli region jest zajęty, tzn. gdy inny proces wykonuje procedurę udo-

stępnianą przez ten region. Zakończenie wykonywania procedury krytycznej (także wstrzymanie procesu ją wykonującego) powoduje, że region staje się wolny, co pozwala wznowić jeden z zawieszonych procesów (określony przez algorytm szeregowania przyjęty w implementacji).

Mechanizmy nadzoru czasowego

Mechanizmy nadzoru czasowego pozwalają programować reakcję na upływ czasu w otaczającym środowisku. Program może wykrywać moment upłynięcia określonego odcinka czasu i synchronizować się do pewnych bezwzględnie określonych chwil czasu. W celu precyzyjnego określenia miejsc w programie, w których może wystąpić przerwanie czasowe wprowadzone zostało pojęcie procesu przerywalnego (time-outable). W języku przewidziano trzy rodzaje instrukcji nadzorów czasowych:

- względny - związany z upłynięciem określonego odcinka czasu (AFTER ...)
- bezwzględny - określający moment czasu (AT ...)
- cykliczny - wyznaczający interwały czasowe (CYCLE ...).

Z pojedynczym procesem może być związanych wiele akcji nadzoru czasowego. Gdy proces jest przerywalny i upłynął co najmniej jeden ze związanych z nim nadzorów czasowych, następuje przerwanie czasowe, które inicjuje przekazanie sterowania do sekwencji akcji stanowiącej właściwą dla niego obsługę (TIMEOUT ...).

Rozszerzenia obiektowe

Oczekiwano, że zalety podejścia obiektowego można również zdyskontować w odniesieniu do projektowania i implementacji dużych systemów oprogramowania, takich jak centrale telefoniczne dla sieci publicznej. W celu weryfikacji tego poglądu w laboratoriach firmy Siemens przeprowadzono pilotowe wdrożenie prototypu centrali. Wykorzystano obiektową wersję języka specyfikacji SDL i środowisko wspomagające programowanie w języku Object-CHILL, w którym dodano mechanizmy definicji klas oraz wyrażania dziedziczenia i polimorfizmu. Metodologia obiektowa została zastosowana do implementacji systemu sygnalizacji abonenckiej ISDN - protokołu DSS1, aplikacji dla systemu sygnalizacji międzycentralowej - ISUP/SS7 oraz podstawowych funkcji zarządzania, utrzymania i ochrony.

Podobne przesłanki motywowały firmy NTT i OKI do wdrożenia obiektowej wersji języka CHILL, która została następnie wykorzystana przy implementacji oprogramowania dla rodziny central D70 [4,8].

Popularność metodologii obiektowej w zastosowaniach informatycznych, wspomniane wcześniej doświadczenia potwierdzające potencjalną atrakcyjność podejścia obiektowego w kontekście zastosowań telekomunikacyjnych oraz opracowanie i opublikowanie zaleceń definiujących obiektową wersję języka SDL (SDL-92) skłoniły ITU-T do podjęcia prac nad wprowadzeniem rozszerzeń obiektowych do języka CHILL.

❖ ❖ ❖

Mimo, że CHILL jest językiem relatywnie „starym" - ma ponad 20 lat - to wytrzymał próbę czasu i jest z powodzeniem stosowany przez największych producentów systemów telekomunikacyjnych. Znalazł zastosowanie przy opracowywaniu oprogramowania dla usług ISDN, IN i systemów komórkowych.

Dzięki odpowiedniej ogólności podstawowych pojęć związanych ze współbieżnością, CHILL nadaje się do projektowania i implementacji systemów rozproszonych. Bliskość koncepcyjna mechanizmów komunikacji, pojęcia procesu i zasad dekompozycji w stosunku do modelu przyjętego w języku specyfikacji SDL pozwala na znaczne zautomatyzowanie przejścia od fazy specyfikacji do implementacji.

Należy także podkreślić, że CHILL, podobnie jak kiedyś Pascal, może doskonale spełnić rolę pomocniczą przy nauczaniu zasad systematycznego i profesjonalnego programowania systemów czasu rzeczywistego.

Literatura:

[1] Brzeziński K., Średniawa M.: CHILL - The Distributed Systems Challenge, Proceedings of the 5th CHILL Conference, Rio de Janeiro, 1990

[2] Gunther W., Wackerbarth G.: Object-Oriented Design of ISDN Call Processing Software, IEEE Communications Magazine, No 4, 1993

[3] Hamonno F., Laeron R., Potin A., Zampoli C.: Software Portage in Distributed Switching Systems, Proceedings of XIIIth International Switching Symposium, ISS'90, Stockholm, 1990

[4] Inafune M., Kikuchi T., Tomizawa N.: An Object-Oriented Software Architecture for Switching Systems: Experiments and Results, Proceedings of XIIIth International Switching Symposium, ISS'90, Stockholm, 1990

[5] ITU: Recommendation Z.200: CCITT High Level Language CHILL, Geneva, 1988, 1993

[6] Jarociński M., Średniawa M.: Wprowadzenie do języka CHILL, WKŁ, Warszawa, 1988

[7] Kimbler K., Jarociński M.: CHILLIT - An Integrated CHILL Environment for PC, Proceedings of the 5th CHILL Conference, Rio de Janeiro, 1990

[8] Maruyama K., Watanabe N., Koyanagi K., Kai T., Tomita S.: A Concurrent Object-Oriented Switching Program in CHILL, IEEE Communications Magazine, No 1, 1991

[9] Rekdal K.: CHILL - the International Standard Language for Telecommunications Programming, URL: http://www.kvatro.no/telecom/chipsy/chill_overview.html

[10] Winkler J.F.H., Diessl G.: Object-CHILL: an Object-Oriented Language for Telecom Applications, Proceedings of XVth International Switching Symposium, ISS'92, Yokohama, 1992

[11] Wirth N.: Toward a Discipline of Real-Time Programming, Communications of the ACM, No.8, 1977.

Marek Średniawa i Krzysztof Brzeziński
są pracownikami Instytutu Telekomunikacji
na Politechnice Warszawskiej

Dodatek: Przykładowy program (źródło: dokumentacja systemu CHILLIT - IT PW)

Przedstawiony program przykładowy, stanowiący fragment oprogramowania komutacyjnego prostego modelu centrali służy praktycznej ilustracji sposobu programowania i samego języka CHILL. W przykładzie posłużono się wersją języka zaimplementowaną w środowisku CHILLIT.

```
/*------------------------ */
/*  Specyfikacja procedur i typów danych   */
/*  wspólnych dla wszystkich modułów       */
/*---------------------------------------- */

/* Część specyfikacyjna modułu Display     */
Display:
MODULE SPEC
/* udostępnienie typów danych i procedur  */
GRANT message_len, message, Devices PERVASIVE;
 GRANT get ! ALL PERVASIVE;
 GRANT put ! ALL PERVASIVE;
 /* Definicja stałej - długość wiadomości */
 SYN message_len  = 32;
 /* Definicje typów danych                */
 NEWMODE
  Devices = SET (inp_, out_, err_,con_, kbd_,
trm_, prn_, com_),
  message = CHARS (message_len);
 /* specyfikacja szkieletów procedur */
 Get_msg:
 PROC ( dev Devices IN, m message OUT );
 END;

 Get_int:
 PROC ( dev Devices IN, i INT OUT );
 END;
        /* .... */
 Put_msg:
 PROC ( dev Devices IN, m message IN);
 END;

END Display;
/*---------------------------------- */
/*Moduł nadzorujący stan łączy abonenckich */
/*---------------------------------- */
/*   Specyfikacja kontekstu modułu Scanner */
CONTEXT
Display: MODULE;
FOR
/*   Część specyfikacyjna modułu Scanner   */
Scanner:
MODULE SPEC
 GRANT on_hook, off_hook, digit,
       start_up PREFIXED line_signals;
 GRANT scanner_id, register,
       max_line PREFIXED;
 SIGNAL on_hook,
        off_hook = ( INT ),
        digit    = ( CHAR('0':'9') ),
        start_up;
  SYN   max_line  = 9;
  DCL
        register ARRAY(0..max_line) INSTANCE
:= [ (ELSE): NULL ],
        scanner_id INSTANCE;
END Scanner;

/*---------------------- */
/* Moduł sterowania centralą telefoniczną */
/*---------------------------------------- */
/* Spec. kontekstu dla modułu Exchange     */
CONTEXTDisplay: MODULE;
Scanner: MODULE;
FOR
/* Część specyfikacyjna modułu Exchange    */
Exchange:
MODULE SPEC
  SEIZE on_hook, off_hook, digit,
        start_up PREFIXED line_signals;
  SEIZE register, scanner_id,
        max_line PREFIXED Scanner;
END;
```

```
/* Część implementacyjna modułu Exchange */
MODULE BODY
 SIGNAL dialled_num = (INT), abort;
  SYN   num_len = 7,
        time_limit = 60;
  DCL   line_no  INT;
/* Procedura odbioru numeru                */
 Receiver:
 PROC (line_no INT) RETURNS (INT)
                    EXCEPTIONS (cancel);
 DCL number INT := 0,
     dummy_id INSTANCE;
  await_digits:
  DO FOR i IN INT(1:num_len);
   RECEIVE CASE BEFORE Time_limit;
    (digit IN d): number := number*10 +
NUM(d) - NUM('0');
    /* Wymuszenie wyjątku cancel_ex */
    (on_hook):    CAUSE cancel_ex;
    (timeout_sg): CAUSE cancel_ex;
   ESAC;
  OD await_digits;
  RETURN number;
 END Receiver;
/* Proces obsługi połączenia               */
 Subscriber:
 PROCESS (line_no INT IN);
   DCL number       INT,
       receiver_id  INSTANCE,
       dummy_id     INSTANCE;
  Connect:
  PROC ( number INT ) EXCEPTIONS (cancel_ex);
    IF number > max_line
    THEN
     Put_int(con_, number);
     Put_msg(con_, " nonexistent");
    ELSIF register( number) /= NULL
    THEN
     Put_int(con_, number);
     Put_msg(con_, " occupied");
    ELSE
     Put_hex(con_, line_no);
     Put_msg(con_, " => ");
     Put_int(con_, number);
    FI;
    Put_eol(con_);
    await_on_hook:
    RECEIVE CASE SET dummy_id;
     /* Wymuszenie wyjątku cancel_ex */
     (on_hook): CAUSE cancel_ex;
      ELSE GOTO await_on_hook;
    ESAC;
   END Connect;
   number := receiver (line_no);
   connect( number );
 END
 /* obsługa wyjątku cancel_ex */
 ON
   (cancel_ex): STOP;
 END Subscriber;
/* ciało modułu exchange  */
 SEND start_up TO scanner_id;
/* główna pętla programu */
 create_connections:
 DO FOR EVER;
  RECEIVE CASE SET scanner_id;
  (off_hook IN line_no):
    /* kreacja procesu Subscriber */
    START Subscriber (line_no)
          SET register(line_no);
  ESAC;
 OD;
END exchange;
```

Język Java i komputer sieciowy...

Jerzy Szyller

Nie da się ukryć, że w mijającym roku Sun Microsystems wraz z kilku silnymi koalicjantami ustawili katapulty najcięższego kalibru ostrzeliwując, wydawałoby się nie do zdobycia, bastiony i flanki zamczyska firmy Microsoft. Język Java pełni rolę nabierającego szybkości i energii tarana, który jeszcze nie uderzył bezpośrednio w bramy królestwa, ale wydaje się, że coraz większa liczba atakujących przyłącza się do dźwigania zgubnej kłody. Następny rok zapowiada się jeszcze ciekawiej - huk będzie li tylko czy wrota wylecą?

Bill Gates najpierw wyśmiał koncepcję komputera sieciowego, jako sprzeczną z bardzo ludzką zasadą chęci posiadania jak najwięcej, aby następnie znacznie stonować swoją pierwszą wypowiedź. Komputer sieciowy jest potrzebny w różnych okolicznościach, ale pozycja tradycyjnego komputera PC z coraz szybszymi procesorami, kilkudziesięcioma Mbajtami pamięci RAM i Gbajtami pamięci dyskowych oraz rozbudowanym oprogramowaniem jest nie do podważenia przez taki „niepoważny" gadżet. Kto ma rację, okaże się wkrótce.

Dzisiaj już ponad 400 producentów oprogramowania zainteresowało się technologią Java, opracowując ponad 450 aplikacji wspomagających zarządzanie przedsiębiorstwem, zwiększających wydajność prac biurowych, umożliwiających pracę w sieci Internet, rozwijanie oprogramowania i usług informatycznych. Takiego ruchu producentów nie można już lekceważyć! W centrum technologii Java znajduje się komputer sieciowy, wykonany na bazie dowolnego procesora. W sieci będzie on akceptował lub inicjował, jako system kliencki, dowolne oprogramowanie napisane w języku Java, zazwyczaj ściągane z serwerów. Pomysłodawcy spodziewają się, że zastosowanie komputerów sieciowych znacznie zwiększy produktywność i obniży koszt systemów sieciowych.

Pierwsze egzemplarze komputera sieciowego pojawią się na początku 1997 roku, zastępując droższe komputery typu PC. Technologia Java będzie miała widoczne sukcesy, zwłaszcza w przypadku dużych sieci różnych organizacji gospodarczych. Oszczędności powinny sięgać milionów dolarów.

Nazwę - **komputer sieciowy (network computer)** -proponuje pomysłodawca, firma Sun Microsystems. Uważam, że wierne tłumaczenie na język polski nie oddaje istoty pomysłu, sugerując pełne możliwości konfiguracji, jaką zazwyczaj posiadają zwykłe komputery - a więc przede wszystkim pamięci dyskowe i urządzenia peryferyjne. Właśnie z założenia niczego takiego w komputerze sieciowym nie ma, a ponadto komputerem sieciowym może być każdy komputer działający w ramach sieci, np. serwer Internetowy! Jakie są inne możliwości? Wymieńmy kilka - **terminal sieciowy, jednostka sieciowa, procesor sieciowy, stacja sieciowa.**

Nazwa - **terminal sieciowy** - również jest myląca, gdyż znacznie zawęża możliwości całego urządzenia do funkcji prezentacyjnych, podczas gdy w rzeczywistości pozwala ono wykonywać korzystając ze swojej pojemnej pamięci operacyjnej złożone oprogramowanie. **Procesor sieciowy** to znowu zbyt wiele lub zbyt mało. Procesor kojarzy się z podstawowym elementem komputera - to za mało, ale również np. z komunikacją - **procesor komunikacyjny**. To zbyt wiele. Jednostka sieciowa kojarzy mi się za bardzo z kolejnictwem i wojskowością. Uważam, że najzręczniejszą nazwą jest - **stacja sieciowa**. Z jednej strony mamy już pojęcie stacji roboczej lub graficznej, które z założenia służą jednemu użytkownikowi w danym momencie, stanowiąc opozycję w stosunku do serwera, natomiast z drugiej, sam producent odchodząc od nazwy ogólnej „network computer" nazywa swój wynalazek, bardzo konkretną nazwą - JavaStation.

Poddając tę propozycję pod dyskusję, mam nadzieję na wywołanie do tablicy specjalistów polskiego oddziału Sun Microsystems. Zapraszam na łamy Informatyki.

Hewlett-Packard wprowadza nowe pojęcie!

Wraz z prezentacją nietypowego urządzenia pojawiły się w języku angielskim dwa nowe słowa: mopying i mopier. Pochodzą one od pierwszych liter trzech wyrazów - Multiple Original Printing. Określenie to oznacza drukowanie wielu oryginałów tego samego dokumentu zamiast jego kopiowania. Nowe słowa „ukuto" na wzór dwóch innych wcześniej istniejących: copying (kopiowanie) i copier (kopiarka). Dlatego w polskim opisie nowego urządzenia posłużyliśmy się podobną metodą słowotwórczą. Zatem określenia „mopowanie" i „mopiarka" prosimy potraktować jedynie jako próbę opisania słowami nowego zjawiska.

Dotychczasowe przyzwyczajenia każą uważać, że do drukowania służą drukarki. Natomiast do powielania używa się fotokopiarek. We współczesnym miejscu pracy funkcje obu urządzeń nakładają się. Coraz więcej pracowników biurowych odkrywa, że szybkie drukarki z wydajnym mechanizmem są w stanie przejąć wiele zadań przypisanych dotąd kopiarkom. Jak wynika z badań amerykańskiej firmy konsultingowej Dataquest, prawie połowa użytkowników drukarek laserowych pomija kopiowanie i drukuje od razu potrzebną im ilość egzemplarzy. Jednak w tradycyjnym pojęciu wielu kierowników administracyjnych nie jest to zwyczajowy sposób powielania dokumentów i dlatego nie akceptują oni mopingu. Mimo coraz większej popularności w wielu wypadkach pozostaje on nienazwanym procesem. Akceptujemy, czy odrzucamy nowe pojęcie? Informatyka zaprasza do dyskusji! (j.sz)

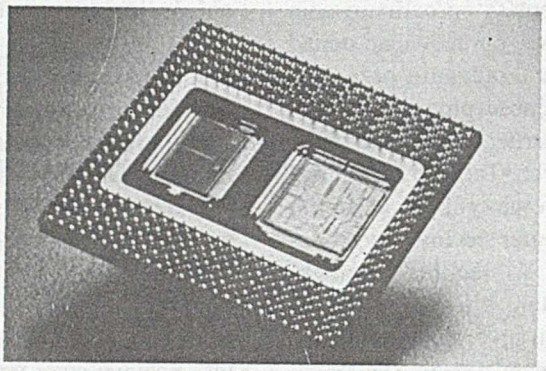
25 lat mikroprocesora czyli po co jemy tę żabę

Jerzy Szych

W końcu lat 60-tych japońska firma Busicom poszukiwała nowych układów logicznych do produkowanych przez siebie programowalnych kalkulatorów. Zamówienie na stworzenie dwunastu nowych układów powierzyła istniejącej od 1968 r. amerykańskiej firmie Intel.

Inżynier Ted Hoff zaprojektował prosty układ wykonujący instrukcje logiczne, zapamiętane przedtem w odrębnym układzie pamięci. Założyciele Intela - Gordon Moore i Robert Noyce dali się przekonać, że układ ten może być wykorzystany nie tylko w kalkulatorach. Za 60 000 dolarów wykupili prawa autorskie (własności intelektualnej) od Busicom i rozpoczęli produkcję układów programowalnych. W ten sposób zaczęła się REWOLUCJA.

W listopadzie 1971 Intel wypuścił na rynek układ 4004, wkrótce potem 8-bitowy 8008. Ostatni krok na tej drodze postawił Ed Roberts w 1974 r. dodając do mikroprocesora klawiaturę i monitor. W ten sposób powstał pierwszy KOMPUTER OSOBISTY ALTAIR.



W 1970 r. sprzedaż całego światowego przemysłu wyniosła około 10 000 komputerów, dziś co dzień instaluje się ich 50 000, zaś łączna liczba pracujących już komputerów przekroczyła 100 milionów. Pierwszy mikroprocesor składał się z 2300 tranzystorów, zaś współczesny procesor Pentium ma ich 5,5 miliona. Obrazową analogię zaproponował Gordon Moore (Intel): *„Gdyby od 1971 r. udoskonalanie samochodów przebiegało w analogicznym tempie, jak rozwój technik komputerowych, dziś Rollce Royce przejeżdżałby milion kilometrów na kilku litrach benzyny"*. On też sformułował prawo potwierdzające się w praktyce już przez ćwierć wieku: ilość tranzystorów zawartych w układzie scalonym podwaja się co 1,5 roku.

Czy zatem nieuchronnie czeka nas dalszy niepohamowany rozwój?

To właśnie pytanie jest dziś przedmiotem analiz, prowadzonych z wielu punktów widzenia. Nie negując technicznych możliwości rozwoju podnosi się kwestie opłacalności tego postępu. Koszt stworzenia nowego układu rośnie o wiele szybciej niż uzyskany w ten sposób wzrost wydajności: zbudowanie nowej fabryki układów scalonych kosztuje około 2 mld dolarów. Koszty modernizacji i rozbudowy systemów PC są tak duże, że najwięksi uczestnicy rynku zbudowali (Sun-Oracle, IBM) lub zamierzają zbudować (nowa koalicja pod wodzą Microsoft) tani i prosty komputer sieciowy.

Inna linia krytyki kieruje się w stronę rezultatów działania systemów komputerowych. Podkreśla się, że współczesne 32-bitowe układy nie są w stanie zrobić właściwie nic więcej niż 8-bitowe, stworzone 25 lat temu, a jedynie robią to o wiele szybciej.

Czy zatem postęp jest znikomy?

Opinie o osiągniętym postępie są zróżnicowane także pod względem socjologicznym. Według ostatnich badań, sponsorowanych przez firmę Motorola, osoby w wieku 16-24 lata i bardziej wykształcone oceniają korzyści z komputeryzacji o wiele wyżej niż osoby starsze i słabiej wykształcone. Kto z nich ma rację?

Jeszcze inny punkt widzenia reprezentują ekonomiści. Zadają oni pytanie, w jakim stopniu komputery przyczyniły się do wzrostu produktywności i bogactwa społeczeństw. Zaskakująca odpowiedź brzmi: w bardzo niewielkim. Wzrost produktywności w 7 najbardziej uprzemysłowionych krajach wynosił w latach 1960 - 73 średnio 3,3% rocznie, zaś w zeszłym roku - zaledwie 0,8%. Podobnie średni dochód realny w USA na osobę spadł w porównaniu do dochodu osiąganego 25 lat temu. Nie epatując się statystyką, która - jak wiadomo - jest w stanie udowodnić dowolną tezę, warto zastanowić się poważnie nad pytaniem o korzyści społeczne.

Ekonomiczny noblista Robert Solow ujmuje to sucho: *„Komputery można zobaczyć wszędzie poza statystykami produktywności"*. Pesymizm ten podziela znany analityk Paul Krugman (profesor Stanford University). W najnowszym numerze miesięcznika Centre Piece, wydawanego przez London School of Economics, stwierdza on, iż ulepszenia techniczne dotyczą tylko niewielkiej sfery: MANIPULOWANIA, PRZETWARZANIA I PRZESYŁANIA informacji w postaci cyfrowej. Dodaje, że *„informacja nie jest jednakże ostatecznym celem gospodarowania. Ludzie pragną większych domów, ładniejszych samochodów, lep-*

szego jedzenia, nie zaś ciągów zer i jedynek. Przekonanie, że ciągi liczb mogą dawać większe domy, ładniejsze samochody i lepsze jedzenie, jest jedynie aktem wiary".

Wprawdzie obecnie dowolny układ scalony ma więcej mocy obliczeniowej niż wszystkie komputery na ziemi na początku lat 50-tych, jednakże ludzki umysł nadal góruje nad nim zdecydowanie.

Czy postęp komputeryzacji nie wpłynie na produktywność społeczeństw?

Profesor Oxfordu i Stanford Paul David opisał niedawno historię prądnicy elektrycznej, gdzie od wynalezienia urządzenia do znaczącego wzrostu produktywności w USA upłynęło 40 lat. Czas ten był potrzebny nie tylko na zainstalowanie urządzeń elektrycznych, lecz także na znalezienie właściwych sposobów ich wykorzystania.

Analogiczne stwierdzenie w odniesieniu do komputeryzacji formułuje Paul Krugman: *„Mamy przed sobą długą drogę dla określenia sposobu, w jaki ta technologia zwróci poniesione koszty przez wywołanie dramatycznego wzrostu produktywności".*

Inni analitycy objawiają znacznie większy optymizm, m.in. wskazując na niedostatki statystyki gospodarczej, gdyż na przykład nie odzwierciedla ona systematycznego wzrostu jakości wyrobów oraz spadku ich cen. Twierdzą też, że nie jest w pełni zasadne badanie produktywności średniej w całej gospodarce, gdyż ogromny postęp występuje głównie w przemysłach związanych z nową technologią.

Dla zrównoważenia pesymizmu Paula Krugmana zacytuję replikę Danny Quaha, dyrektora programu Badania Narodowej Produktywności w Centrum Badania Produktywności Gospodarczej USA: *„Być może nie najemy się bajtami pamięci, ale podobnie nie możemy jeść stali czy węgla, a mimo to przez dziesięciolecia oznaką silnej gospodarki była zdolność do ich produkcji. Były one uważane za środek poprawiający jakość życia i bez wątpienia tak właśnie było. To samo stanie się z produkcją zer i jedynek".*

Jerzy Szych
jest pracownikiem ICL Poland

O mitologii komputeryzacji – moje trzy grosze

Stefan Sokołowski

Z dużym zainteresowaniem przeczytałem artykuł [1] Ryszarda Łączkowskiego. Z wieloma jego tezami, dotyczącymi szkodliwości bezkrytycznego wypełniania komputerami wszystkich zakamarków naszego życia społecznego, zgadzam się w pełni. Podzielam również przekonanie, że *„komputer jest urządzeniem [...] wspaniałym, ale nigdy – – cudownym"*. Dodałbym do tego, że w nastawieniu społecznym do komputerów kryje się zagrożenie dla nas, informatyków: po fali przesadnego uwielbienia przyjść może równie przesadne rozczarowanie i wtedy pozostaniemy bez pracy. Już dość dawno usiłowałem (z miernym skutkiem) zwrócić uwagę społeczną na zawodność analiz komputerowych – por. [2].

Jednak z rzetelności badawczej czuję się zmuszony skorygować pewne niezbyt precyzyjne stwierdzenia prof. Łączkowskiego. Mam nadzieję, że moich trzech groszy Czytelnik nie potraktuje jako totalnej krytyki artykułu. We wszystkich omawianych tam sprawach, o których poniżej nie wspominam, zgadzam się z autorem. A w niektórych zgadzam się, ale uważam, że należy inaczej argumentować.

Sztuczna inteligencja

Mam wrażenie, że dla tego kierunku badań oraz dla samego terminu jest pan profesor niesprawiedliwy. „Sztuczna inteligencja" to bynajmniej nie to samo co „program komputerowy"; to jest cała dziedzina wiedzy. Ludzie różnie definiują jej zakres; ja powiedziałbym , że jest to nauka o tym, w jakich inteligentnych czynnościach człowieka można obyć się bez inteligencji. Typowym przykładem są szachy. Niegdyś traktowane były jako coś w rodzaju testu na myślenie w ogóle, a teraz przeciętny użytkownik komputera przegrywa z bezmyślnym urządzeniem stojącym na jego biurku (chociaż mistrz szachowy wygrywa). A więc mimo, iż w tej grze „*człowiek*" musi opierać się na inteligencji i intuicji, komputer może mu (prawie) dorównać, wykonując szybko i bezbłędnie odpowiedni algorytm. Badania w dziedzinie sztucznej inteligencji doprowadziły do takiej właśnie detronizacji tej królewskiej gry.

Szachy są klasycznym wyzwaniem dla sztucznej inteligencji i obszarem, na którym osiągnięto najbardziej widowiskowy sukces. Inną klasyczną dziedziną jest rozpoznawanie obrazów. Niegdyś uważano za oczywiste, że kształty rozpoznawać może tylko mózg wytworzony przez naturę. Dzisiaj można sortować listy na poczcie w oparciu o ręcznie napisany kod oraz automatycznie anali-

zować tysiące zdjęć satelitarnych lub setki fotografii nowotworów, dostarczonych przez skaner. I znów udało się zastąpić ludzkie myślenie i intuicję algorytmem. A że ten algorytm działa bardziej zawodnie niż człowiek? Ale za to o ile szybciej i taniej! Właśnie takie bezmyślnie działające algorytmy, odciążające ludzką inteligencję do bardziej godnych celów, są przedmiotem badań sztucznej inteligencji.

Każdą dziedzinę wiedzy charakteryzuje zakres tematyczny oraz stosowana metodologia badawcza. Omówiłem mniej więcej zakres tematyczny sztucznej inteligencji, a z metodologią mam niestety kłopot. Alan Bundy skarżył się kiedyś, że każda udana metodologia wypracowana przez badaczy od sztucznej inteligencji, staje się zaczątkiem nowej dziedziny wiedzy, natychmiast po osiągnięciu dojrzałości i w ten sposób sztuczna inteligencja ciągle nie ma własnych, dobrze ugruntowanych metod badawczych. To jest trochę tak, jak z filozofią, która niegdyś była synonimem nauki w ogóle, a z której potem wyłaniały się kolejne nauki i w końcu trudno powiedzieć, co jej pozostało na własność.

Nikomu nie zależy na zbudowaniu sztucznego mózgu, zastępującego ludzkie myślenie idealnie i we *wszystkich* jego funkcjach.

Termin „sztuczna inteligencja'' był i nadal jest krytykowany z powodów wymienionych w artykule i jeszcze z wielu innych. Ryszard Łączkowski zestawia go z wyrażeniem „układ inteligentny'' i twierdzi, że wzbudzają one sprzeciwy uczuciowe oraz *„budzą wyraźną niechęć do ich stosowania"*. Jeśli tak, to mamy odmiennie ukształtowaną sferę emocji; termin „sztuczna inteligencja'' niczego takiego nie wzbudza ani we mnie ani w wielu innych ludziach, czego świadectwem jest to, że się tak bardzo rozpowszechnił i że od wielu lat nie znaleziono dla niego żadnej alternatywy (nie mogę się wypowiedzieć na temat „układów inteligentnych'', bo nigdy o nich nie słyszałem; przypuszczam, że jest to slogan reklamowy jakiejś firmy). Osobom mylącym inteligencję ze sztuczną inteligencją należy oczywiście bez przerwy powtarzać, że jedno do drugiego ma się tak, jak rekord sportowy do record'u w Pascalu lub jak koń do koniaku.

A co mamy sądzić o meczu, w którym Kasparow pokonał superkomputer Deep Blue z wynikiem 4:2? Ano tylko to, że *mój* wynik przeciwko Kasparowowi byłby bez wątpienia gorszy; czy na tej podstawie pan profesor odmówi mi posiadania inteligencji?

Przewidywalność działania komputera

Cytuję z artykułu: *„Tylko niepoprawni fantaści mogą twierdzić, że z komputerów będzie można w przyszłości budować systemy inteligentne, które zastąpią twórcze myślenie człowieka. Jest to niemożliwe, ponieważ mózg ludzki pracuje na zasadzie skojarzeń, natomiast komputer działa według metody adresowej. Wynik ludzkich skojarzeń nie zawsze jest jednoznaczny i przewidywalny, w odróżnieniu od rezultatów analizy komputerowej, która jest zawsze zdeterminowana i obarczona błędem możliwym do oceny. Myślenie ludzkie potrafi przystosować się do nowych sytuacji i zadań, natomiast komputer może realizować tylko procedury przewidziane przez programistę."*

Nikomu nie zależy na zbudowaniu sztucznego mózgu zastępującego ludzkie myślenie idealnie i we *wszystkich* jego funkcjach. Będziemy chcieli zastąpić tylko niektóre elementy myślenia człowieka, np. liczenie, wyszukiwanie optymalnego rozwiązania itp. Konstruktorom samolotów też nie zależy na zastąpieniu ptaka w jego funkcji wysiadywania jajek lub wydziobywania pędraków ze ścierniska, a za to koncentrują się na lataniu. To, że zasady lotu ptaka i samolotu są różne, też im specjalnie nie przeszkadza. A więc jestem takim niepoprawnym fantastą, który uważa, że mimo dramatycznych różnic w zasadach działania, komputery przyszłości będą przejmowały od ludzi niektóre zadania, które bywają nazywane „inteligentnymi'' lub „twórczymi''. Będą temu towarzyszyły stałe zmiany ludzkiego rozumienia pojęcia „twórczość'' lub „inteligencja''. Na poparcie powyższego przewidywania mam tylko jeden argument, mianowicie... że tak się dzieje już od dobrych trzydziestu lat.

W pomieszanie wprowadza mnie utożsamienie w cytowanym fragmencie inteligencji z nieprzewidywalnością. Mimo wszystko przypuszczam, że swoim studentom stawia pan profesor dobre oceny za odpowiedzi przewidywalne, a te nieprzewidywalne trafiają do pańskiej kolekcji anegdotek egzaminacyjnych i śmieje się pan z nich razem z kolegami. W dodatku z przewidywalnością wyników działań systemów komputerowych nie jest tak dobrze, jak wskazywałby na to powyższy cytat. Zastanówmy się, co to znaczy „przewidywalność''. Pierwszy naiwny sposób rozumienia przewidywalności, to pogląd, że programista z góry *wie*, co wyświetli komputer po zakończeniu obliczeń. No tak, ale jeśli wie, to po cóż w ogóle uruchamia program? Używamy komputera, żeby dowiedzieć się czegoś nowego. W tym sensie wynik działania komputera jest prawie zawsze nieprzewidywalny. Możemy podać trochę bardziej wyrafinowany sposób rozumienia tego terminu: przewidywalność działania komputera polega na tym, że istnieje *teoretyczna* możliwość przewidzenia wyniku, nawet, jeśli leży ona (ta możliwość) poza zasięgem pojedynczego programisty. Teraz z kolei będziemy mieli trudność z rozziewem między teorią a praktyką. Powiedzmy, że programista nie wie, co za chwilę wyświetli komputer, ale jest świadom, że zespół 1000 fachowców najwyższej światowej rangi mógłby to w zasadzie przewidzieć, gdyby poświęcił na to 1000 lat pracy. Czy tego rodzaju przewidywalność wyniku różni się istotnie od jego nieprzewidywalności? Użytkownicy komputerów, a w jeszcze większym stopniu użytkownicy sieci komputerowych, są zwykle boleśnie świadomi trudności w takim przewidywaniu.

Programiści starają się panować nad oprogramowaniem, ale nieprzewidywalność wkrada się niechciana; a to jakieś dwie składowe systemu nie chcą ze sobą współpracować,

a to nieznaczne błędy popełnione przez różne zespoły nie znających się ludzi kumulują się do jakiegoś bezsensownego działania mojego programu... Każdy z nas potrafi zacytować z własnego doświadczenia wiele takich przypadków. Często poszukiwanie przyczyn takiego, a nie innego zachowania systemu komputerów jest po prostu nieopłacalne; poprawiamy coś „na wyczucie", żeby uniknąć najbardziej oczywistych bezsensów; uczymy się żyć z nieco mniejszymi nieprawidłościami. I na tej niepewnej podstawie budujemy następną warstwę oprogramowania. W wyniku takiego procesu stopniowo powstaje niezwykle złożony organizm, którego zachowania nie umiemy dokładnie opisać.

Ostatecznie, kiedy skarżymy się fachowcowi, że zainstalowany przez niego system nie spełnia oczekiwań, fachowiec działa nieco po omacku starając się zaradzić problemowi. Wymienia jakiś fragment oprogramowania na nowszą wersję, bo gdzieś obiło mu się o uszy, że ze starą bywają problemy. Albo zmienia jakieś parametry, bo ma niejasne wrażenie, że mogą mieć coś wspólnego z problemem. Albo doradza wymianę płyty głównej, bo jakiś driver nie chce z nią współpracować; a czy będzie współpracował z nową? — chyba będzie, ale trudno to z góry przewidzieć. Fachowiec rozumie system komputerowy tak, jak psycholog rozumie dziecko: wystarczająco, żeby w typowych sytuacjach zaradzić złu, ale gdzie mu tam do pełnej przewidywalności zachowania. A więc nie w tym tkwi główna różnica pomiędzy myśleniem ludzkim a „myśleniem" komputera.

Opłakane skutki tajności

Jak świat światem, etos człowieka nauki i człowieka interesu różniły się od siebie zasadniczo. Jedną z różnic był stosunek do jawności. Naukowcy są od tego, żeby odkrywać i ogłaszać całemu światu, podczas gdy podstawą etyki biznesmena jest dyskrecja. W miarę, jak informatyka przestała być wyłączną domeną naukowców, pojawiły się problemy praw autorskich oraz niejawność szczegółów realizacji. Profesor Łączkowski słusznie obawia się powierzania odpowiedzialności za działania ważne dla człowieka programom, których tekst źródłowy nie został ujawniony, a więc nie ma nawet teoretycznej możliwości sprawdzenia jego poprawności.

Jednak i tutaj należy uczynić kilka zastrzeżeń. Po pierwsze - utajnieniu podlegają konkretne realizacje, a nie algorytmy. Algorytmy są znane powszechnie, a twórcy oprogramowania na ogół nie czynią tajemnicy z tego, *jaki* algorytm zastosowali. Możemy mieć oczywiście wątpliwości, czy zastosowali go poprawnie, ale upewnienie się o tym byłoby trudne, nawet gdybyśmy mieli szanse obejrzeć źródła programu.

Tekst programu to nie romans i nikt go nie czyta do poduszki. Dawniej zaglądało się do źródeł wtedy, gdy program zachowywał się niepoprawnie; obecnie użytkownik spotyka się raczej z niepoprawnym zachowaniem całego wielkiego systemu programów. Podstawową koniecznością jest w takim wypadku ustalenie, który element systemu działa wadliwie - często jest to zadanie bardzo trudne. A gdy znajdzie się już tę błędną składową, to się ją raczej wymienia (przez sprowadzenie lub wygenerowanie nowej), niż naprawia poprawiając jej kod źródłowy. Dlatego skutki jego niedostępności są mniej dramatyczne, niż możnaby oczekiwać. Zresztą w technologiach nieinformatycznych obserwujemy podobne zjawisko: mechanik samochodowy coraz rzadziej musi rozumieć szczegóły funkcjonowania elementów mechanizmu, a za to powinien sprawnie diagnozować, który z nich źle działa i posiadać w magazynie część zapasową.

Istnieje zresztą wielki i rozbudowany system programów, do których publicznie dostępne są starannie skomentowane teksty źródłowe oraz dokumentacja: mam na myśli projekt GNU. Każdy może te źródła ściągnąć z Internetu, zmienić w dowolny sposób, skompilować, uruchomić i rozprowadzać, pod warunkiem, że nie utajni poprawionego przez siebie kodu źródłowego. Jestem użytkownikiem Linuxa, Emacsa, kompilatora gcc i kilku innych produktów GNU i mam wysokie mniemanie o ich jakości. Myślę, że jawność źródeł jest w tym wypadku jednym z powodów tej wysokiej jakości, chociaż związek między jednym a drugim jest raczej natury psychologicznej i społecznej, niż ściśle technicznej.

Jednak nie wydaje mi się, żeby ta jawność *rozwiązywała* problem błędów w oprogramowaniu. I tak nie jestem w stanie przeczytać nawet drobnej części tekstów tych programów, do których działania mam zastrzeżenia. Cenię sobie natomiast fakt, że mając w ręku źródła mogę wyprodukować działające binaria na dowolną konfigurację komputerową.

No tak, ale to już nie jest problem jawności, tylko „przenoszalności" programów.

Cytowane artykuły:

[1] Łączkowski Ryszard, Fetyszyzm i mitologia komputeryzacji. *Informatyka* 10, str. 40—44, 1996

[2] Sokołowski Stefan, Komputerowa entomologia, czyli życie pośród pluskiew. *Problemy* 501, str. 9—13, 1988. Streszczenie jest dostępne w Internecie pod adresem http://www.ipipan.gda.pl/~stefan/Papers/popular.html.

Stefan Sokołowski jest pracownikiem Polskiej Akademii Nauk, Instytutu Podstaw Informatyki, filia gdańska w Sopocie
e-mail: s.sokolowski@ipipan.gda.pl

Sprostowanie

Redakcja przeprasza Czytelników i Autora artykułu „Charakterystyka języka OCCAM", zamieszczonego w numerze 11/96 naszego czasopisma, za błędy w wydrukach programów.

Wznowienie

Kilka lat temu pisywałem do „Informatyki" krótkie felietony - zBITki myśli i spostrzeżeń, a czasem były to zbytki z aktualnych wtedy wydarzeń informatycznych. Teraz postanowiłem powrócić do tych felietonów, mając nadzieję przekonać niektórych do aktywniejszych działań w sferze ogólnie pojętej informatyki. A może w ten sposób inni przekonają mnie, że nie warto albo trzeba zupełnie inaczej spojrzeć na tylko z pozoru jednoznaczne fakty i zjawiska.

Pisząc te felietony, do czasu aż się wam, drodzy Czytelnicy, może znudzą, lub gdy stwierdzę, że nie mam nic ciekawego do powiedzenia, muszę wyraźnie złożyć zastrzeżenie podane na końcu tego tekstu.

Argumentem za podjęciem tej pisaninki jest próba redakcji „Informatyki" w wykreowaniu nowej formuły pisma. Pośród kilkunastu już pism teleinformatycznych - najstarsze z nich - „Informatyka" (dawne Maszyny Matematyczne) nieco zastygła w już przestarzałej formie i treści. A przecież profesjonalna kadra informatyczna w Polsce nie ma własnego pisma, a może ma - które? Tak, tylko kim jest ta profesjonalna kadra? A może nie istnieje?

Jak profesjonalna, to zacznijmy od definicji. W moim pojęciu, profesjonalną kadrą informatyczną są ci wszyscy, którzy bądź ukończyli studia o profilu informatycznym lub podobnym, bądź też mając nieco inne wykształcenie rzetelnie przez kilka lat byli związani zawodowo z informatyką. Dodatkowo starsi powinni się wykazać intensywnym dokształcaniem oraz dodatkowymi szkoleniami z zakresu informatyki. Najwyższy czas, aby ta lub nieco zmodyfikowana definicja zaczęła funkcjonować w powszechnej świadomości. Dla naszego wspólnego interesu i lepszej przyszłości powinniśmy zacząć szanować nasz zawód, tak jak to czynią adwokaci, lekarze, ba - nawet doradcy finansowi, nie dopuszczając znachorów informatycznych, których niestety jest wielu; stanowczo zbyt wielu.

Było już kilka prób regulacji tej sytuacji. Propozycja stopni specjalizacyjnych była niezła i szkoda, że nie udało się jej zrealizować. Być może trzeba zacząć od powoływania rzeczoznawców w dziedzinie informatyki (jest to bardzo pojemne pojęcie), a z czasem zacząć wymagać, aby każdy projekt oraz realizacja systemu była podpisana przez uprawnionego do tego specjalistę (oczywiście z odpowiedzialnością za skutki tego działania, ale też za odpowiednio wysokie wynagrodzenie). Tyle na temat - na razie.

„Informatyka" tylko dla profesjonalistów - tak, ale również powinna być ona zrozumiała dla naszych szefów - decydujących o środkach na informatyzację. Naszym celem powinno być stworzenie pisma tworzonego przez profesjonalistów dla kadr kierowniczych, którzy niekoniecznie muszą nawet lubić informatykę, ale powinni rozumieć i polubić „Informatykę". Dlatego też zadaniem każdego informatyka powinno być wyszukanie w „Informatyce" pożytecznego dla swojego szefa artykułu i zareklamowanie go (mam nadzieję, że redakcja będzie w stanie zamieszczać takie artykuły).

Dla tych, co mnie nie znają - kilka słów o sobie - lepiej wiedzieć, z kim się ma do czynienia. Prezes Polskiej Izby Informatyki i Telekomunikacji, Właściciel firmy konsultingowej „Iszkowski & ska". Członek Rady Koordynacyjnej ds. Teleinformatyki przy RM URM. Poprzednio: Wiceprezes i Dyrektor Generalny w 2SI, Marketing Manager w Digital Equipment Polska, Konsultant dla Oracle Corp. Projektant systemów CAD w Tessel AB i przez 18 lat adiunkt oraz asystent w Instytucie Informatyki Politechniki Warszawskiej. Autor kilkunastu artykułów z analizy polskiego rynku teleinformatycznego, popularnonaukowych i naukowych. Współautor książek: „Programowanie współbieżne", „Implementacja systemów operacyjnych", „Struktury i typy danych", Programowanie w Fortranie", „BASIC", itp. Doktor nauk technicznych i inżynier (specjalność maszyny matematyczne) - absolwent Wydziału Elektroniki Politechniki Warszawskiej. Członek Polskiego Towarzystwa Informatycznego, itp. Przewodniczący Komitetu Organizacyjnego I Kongresu Informatyki Polskiej w 1994 r. Podwójny laureat nagrody Infostar.

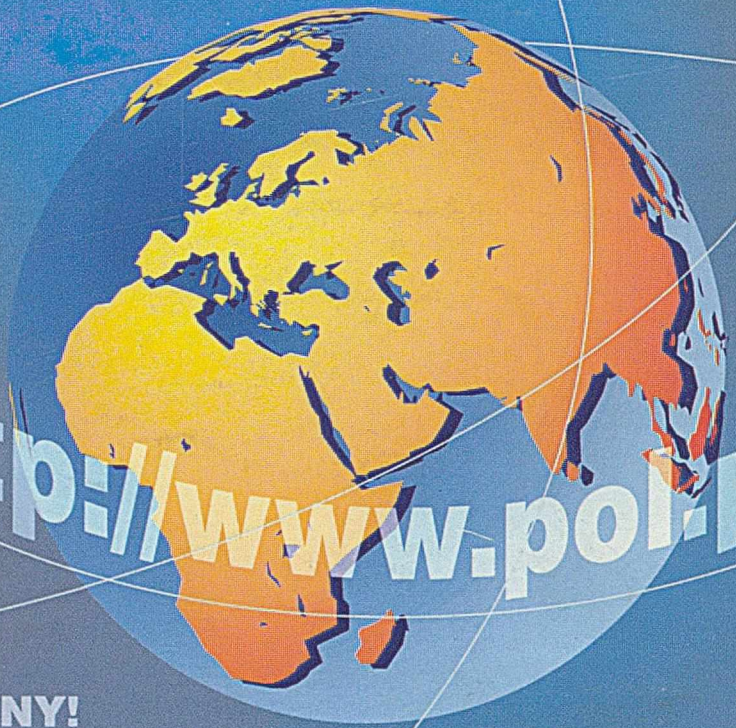
Zastrzeżenie: opinie i stwierdzenia zawarte w felietonach „zBITki" są wyłącznie opiniami własnymi autora i nie należy ich utożsamiać z żadną organizacją, instytucją czy firmą, z którą był, jest lub będzie związany autor.

Wacław Iszkowski

JUNISOFTEX
20 LAT DOŚWIADCZEŃ WDROŻENIOWYCH
Perfect-Ekspert
Sieciowy i zintegrowany system informatyczny pracujący na platformach PC i AS/400, głównie dla dużych firm produkcyjnych i handlowych.
Przystosowany do elektronicznej wymiany dokumentów (EDI-zamówienia, faktury, WZ itp.).
Wyposażony w system ochrony danych generowanych wielowalutowości oraz rachunkowości zarządczej.
wewnątrz i na zewnątrz przedsiębiorstwa.
Analizy Marketingowe
System Informacji Kierownictwa
System Finansowo Kosztowy
System Gospodarki Materiałowej
System Obrotu Towarowego i Fakturowania
System Przygotowania, Planowania i Rozliczania Produkcji
System Środków Trwałych i Wyposażenia
System Kadry-Płace
Monitorowanie Procesów Produkcji, Kontrola Jakości
Inżynieria Wiedzy Systemy Ekspertowe
Inne Systemy Ewidencyjne
Inteligentny System Analiz Finansowych
Elektrownia RYBNIK
Zakłady Rowerowe "ROMET"
Krywałd "ERG" S.A.
Huta Łaziska
Huta Bankowa
Huta 1-go Maja
Huta Szkła Kunice
Zakłady Metalurgiczne Trzebinia
Walcownia Metali "Dziedzice"
Zakłady Górniczo - Hutnicze "BOLESŁAW"
Zakłady Gumowe Górnictwa
Fabryka Osi Napędowych MOSTOSTAL
Elektromontaż 3
POLIFARB Cieszyn S.A.
POLAM Kostuchna
PKM Katowice
WEDEL S.A.
Instytut Informatyki Teoretycznej i Stosowanej
Centrum Naukowo Techniczne Kolejnictwa
Instytut Spawalnictwa
Oferujemy kompleksowe usługi w zakresie:
- doradztwa organizacyjnego, merytorycznego i technicznego
- organizacji, projektowania i wykonawstwa innych systemów informatycznych
- wdrażania, konserwacji i modernizacji systemów informatycznych
- projektowania, wykonawstwa dowolnego typu sieci komputerowych, z uwzględnieniem już istniejących rozwiązań komputerowego i sieciowego
- dostawę sprzętu komputerowego i sieciowego
- szkoleń użytkowników w zakresie operatora i administratora systemu AS/400, jak również obsługi pakietów programowych PC Support 400 lub Client Acces /400 przeznaczonych do współpracy z PC
PRZEDSIĘBIORSTWO PRODUKCYJNO-USŁUGOWE INNOWACJI TECHNICZNYCH JUNISOFTEX
BIURO TECHNICZNO-HANDLOWE, GLIWICE, ul. Kościuszki 16, box 507, tel./fax: (0-32) 1306931, 1306932, 1306933, 1306934, telex 0316220

Polska OnLine

INTERNET



http://www.pol.pl

JESTEŚ AKTYWNY!
BĄDŹ INTERAKTYWNY!

Polska OnLine Sp. z o.o. 01-756 Warszawa, ul.Przasnyska 6,
tel. 639 85 45, fax 663 52 81, e-mail: info@pol.pl